SOCIO-SEMANTIC CONVERSATIONAL INFORMATION ACCESS

A Thesis
Presented to
The Academic Faculty

by

Saurav Sahay

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
College of Computing

Georgia Institute of Technology
December 2011

SOCIO-SEMANTIC CONVERSATIONAL INFORMATION ACCESS

Approved by:

Professor Ashwin Ram, Advisor
College of Computing
Georgia Institute of Technology

Professor Eugene Agichtein
Mathematics and Computer Science Department
Emory University

Professor Shamkant B. Navathe
College of Computing
Georgia Institute of Technology

Professor John Stasko
College of Computing
Georgia Institute of Technology

Professor Mark Braunstein
College of Computing
Georgia Institute of Technology

Date Approved: xxx

To maa, papa and dadu

"What information consumes is rather obvious: it consumes the attention of its recipients. Hence a wealth of information creates a poverty of attention and a need to allocate that attention efficiently among the overabundance of information sources that might consume it"

Herbert Simon. (1971), "Designing Organizations for an Information-Rich World", (p. 40-41).

ACKNOWLEDGEMENTS

The journey through a PhD thesis and an academic life is an experience of a lifetime that I've cherished and enjoyed dearly. My deepest gratitude goes to my advisor Ashwin whose vision, motivation and intellectual guidance has inspired me through this journey. Working with Ashwin has given me so much of exposure and experience in a few years that I feel is most valuable quality of an independent thinker, doer and a researcher. I heartily thank Dr. Navathe who has supported me from the very beginning of my graduate life here and has given me the right advice and training and believed in my abilities as someone who can make valuable contributions in discussions and meetings and advise junior students. I am also grateful to Eugene with whom I closely worked initially on another project and learnt from his deep insights and thoroughness. Talking to him would trigger multiple simultaneous spreading activations in my mind and would result in great learning! I am also greatly thankful to Mark who exposed me to the vast field of Health Informatics and Healthcare with deep knowledge and practical suggestions throughout the cobot project. My gratitude also goes to Dr. John Stasko who accepted to be part of my thesis committee.

I would like to thank my co-workers, lab-mates and friends with whom I've worked, learnt and have had fun together during the course of many different projects here. I thank Anushree for being the CEO of cobot project for some time and dreaming it along with me for it's success. My regards and gratitude also goes to Alejandro who brought in tremendous technological experience in this project. Many students have worked on cobot project and I would like to thank each one of them: Abbas, Hrishikesh, Stephanie, and Bharat for their contributions in the project.

Another important part of my graduate life has been my internships at IBM Research where I've learnt lots of great technologies from researchers and many mentors there. I thank Dr. Eric Mueller, Dr. Sougata Mukherjea, Dr. Sugato Bagchi and Dr. Anthony Levas, Dr. Bran Boguraev and Dr. James Cooper who have mentored me during my internships and given me tremendous practical guidance.

Being an international student, living with roommates, one's friends become one's family. I thank my friends and past roommates Rohan, Manav, Rohit, Sudeep, Vikrant, Prashant, Vyomkesh, Deepa bhabhi and Anuja bhabhi for the amusements, parties and fun we've had together over these years. I am also thankful to my friend Subharthi for intellectual discussions at demand time any hour of the day or night with him.

I am fortunate to have had my childhood friend and now family Nishant at my side during the good and bad days I've had here. He has supported me, encouraged me and reprimanded me for staying focussed in life. I am thankful to Chhoti, Soni, Maa and Papa to have had faith in me and provided love and support throughout this journey.

Lastly, I thank all those who have supported me during the completion of my graduate school.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

This thesis lies broadly in the field of intelligent information access, primarily at the intersection of language processing, user modeling and web based socio-informatics systems. The main contributions revolve around developing this integrated conversational recommendation framework, combining data and information models with community network and interactions to leverage multi-modal information access. This work has been influenced by a number of fields such as Information Retrieval and Extraction, Case based Reasoning, User Modeling and Adaptation, and Socio-Technical Systems.

We have developed a real time conversational information access community agent that leverages the community knowledge by pushing relevant recommendations to users of the community. The recommendations are delivered in the form of web resources, past conversation and people to connect to. The information agent (cobot, for community/ collaborative bot) monitors the community conversations, and is 'aware' of users' preferences by implicitly capturing their short term and long term knowledge models from conversations. The agent leverages from health and medical domain knowledge to extract concepts, associations and relationships between concepts, formulates queries for semantic search and ultimately provides socio-semantic recommendations in the conversation after applying various relevance filters to the candidate results. The agent also takes into account users' verbal intentions in conversations while making recommendation decision.

One of the goals of this thesis is to develop an innovative approach to delivering

relevant information using a combination of social networking, information aggregation, semantic search and recommendation techniques. The idea is to facilitate timely and relevant social information access by mixing past community specific conversational knowledge and web information access to recommend and connect users with relevant information.

With an explosion in proliferation of user-generated content, the productivity of search is decreasing and quality of readily available online content is deteriorating. There is an increasing need for intelligent assistants that can understand user interactions in the social context for better addressing the problem solving needs of the user. Cobot models user utterances in conversations to proactively target the community for exchange of questions and answers in conversations. We envision a system that encourages user engagement and participation by prompting questions and asking to suggest answers based on user's knowledge and activity levels.

One problem with social information systems is the noise-signal ratio. This ratio becomes high due to informal nature of the language in conversations in communities which hinders relevant recommendations. One solution is to normalize the community conversations to extract meaningful representations using conceptual knowledge coming from socially generated tags or knowledge from an ontology. This underlying conceptual base for consumption and participation drives internal knowledge representation for the socio-semantic system.

Language and interaction creates usable memories, useful for making decisions about what actions to take and what information to retain. Cobot leverages these interactions to maintain users' episodic and long term semantic models. The agent analyzes these memory structures to match and recommend users in conversations by matching with the contextual information need. The social feedback on the recommendations is registered in the system for the algorithms to promote community preferred, contextually relevant resources.

The nodes of the semantic memory are frequent concepts extracted from user's interactions. The concepts are connected with associations that develop when concepts co-occur frequently. Over a period of time when the user participates in more interactions, new concepts are added to the semantic memory. Different conversational facets are matched with episodic memories and a spreading activation search on the semantic net is performed for generating the top candidate user recommendations for the conversation. The activation is spread to the neighboring nodes proportional to the weight of each connecting association in the semantic net. There are several parameters in the system that can be learnt based on activity of users. Parameters for episodic memory window size, semantic memory learning and unlearning rates, concept co-occurrences and feedback strengths for associations are initially set heuristically and can be fine-tuned to suit individual users.

The tying themes in this thesis revolve around informational and social aspects of a unified information access architecture that integrates semantic extraction and indexing with user modeling and recommendations.

CHAPTER I

INTRODUCTION

"Solving the problem of bringing relevant information to people using conversational information access"

1.1 Overview

Online user generated content is proliferating at a never before rapid pace on the web. The knowledge explosion by this proliferation has continued to outpace technological innovation in efficient information access technologies. With increasing amount of data and noise, ease of high quality information access has become difficult. This has resulted in users spending more time to sift through lots of information, feeling of information overload, disengagement and lack of attention. The trend is towards development of better Web 3.0 tools, Semantic Web Services, Recommendation agents, etc. that try to engage the users by providing memory aids, personalization and feedback based active engagement, semantic understanding and reformulation of user's information need to retrieve high quality content, collaborative and social information seeking, etc. Recommendation technologies are replacing search technologies that include general methods for information access like browsing pages on information portals and social media, and querying search engines or finding information in forums and message boards.

We define Socio-semantic access as a method of unified information access that involves a seemless integration of social entites with semantic entities to provide relevant information to people either through people or through resources identified using knowledge elicitation using semantic means.

In this thesis, we introduce a natural language Socio-semantic Conversational Information Access method to facilitate agent assisted, socially filtered, semantically analyzed information access as a solution for interactive, collaborative and dynamic information access. Conversational Information Access (CIA) is an interactive and collaborative information seeking interaction. The participants in this interaction engage in a conversation aided with an intelligent information agent (Cobot) that provides contextually relevant recommendations and connects relevant users together. This collaborative CIA aims to engage users and raise awareness of relevant information, and improve the search and discoverability of relevant information. This thesis takes a knowledge centric and domain guided approach to information access. We have incorporated knowledge from the health and medical domain by creating large semantic dictionaries extracted from biomedical ontologies. We also show that cheaply available, domain specific, socially generated tags are also effective for domain specific conversational recommendations.

Conversational Information Access leverages the search and discovery process by integrating web information retrieval along with the social interactions. A typical Google or Yahoo Answers experience is solitary and repetitive, while the conversational approach is collaborative and dynamic and aims to be engaging. Cobot is an intelligent agent that monitors community interactions, uses domain specific knowledge for finding recommendations and brings relevant information to users by augmenting the conversations. Cobot's 'conversation engine' monitors user conversations with other users in the community and provides recommendations based on the conversation to the participants. Cobot's 'community engine' models conversations to capture user-user and user-information interactions. Cobot leverages collaborative and conversational information access by harnessing the collective intelligence of users and information from the web.

This research explores methods that combine passive web search with interactive

social search for efficient information access. The agent embedded conversational information access system leverages from domain specific knowledge and personal experiences of users and aims to increase the usability of this information system by bringing these together. The agent leverages community interactions by building socio-semantic conversation models to capture user-user and user-information interactions.

The contributions of this thesis also include an innovative blended platform that combines community information seeking with web search and results in more effective information access through a natural experience of conversations, collaboration and socio-semantic recommendations. Instead of performing a solitary search for information by typing words into a search engine, people engage in meaningful discussions with others having similar needs and interests and leverage each others' information. The Cobot community includes information seekers and providers, who participate in the community for information, learning and education. Community members provide and consume information in the form of user-generated personal experiences and conversations. The Cobot platform connects users together for conversations, and provides contextually relevant recommendations based on ongoing conversations. The recommendations in a real time conversational environment provide users with convenient access to reliable and contextually useful information through a natural experience.

With the huge number and types of sources available on the web, searching for information is continuously becoming more difficult. A knowledge worker has to spend inordinate amounts of time researching for information on the internet. There is not enough time to focus on individual search contexts and sift through pages of results. The problem in complex domains such as healthcare may be even more acute than that in general search due to the degree of sensitivity associated with the domain information. It is also more difficult to formulate an accurate search query, and the

generic information returned by search engines is unlikely to pertain to your specific problem. The search query is also not unique and a lot of people might already have the answers to common problems. Search innovation and Technology should make information more easily and quickly accessible for users. Bringing in relevant people to answer one's specific problems as well as intelligent agent based recommendations to augment the collective knowledge addresses the market need for good, relevant and meaningful information. Technology should enable the power to collaborate on individual tasks and provide help to solve them together.

1.2 Market Need and Applications

In this thesis, we primarily develop domain specific conversational recommender, leveraged on top of a large medical knowledgebase with millions of terms and their semantic types. The medical knowledgebase called UMLS (Unified Medical Language System) includes concepts belonging to categories such as diseases, drugs, findings, treatments, etc. The healthcare industry is moving towards a more consumer-centric focus with skyrocketing healthcare cost, the aging population, the increasing lack of doctors, and the improved accessibility of medical information. Health awareness amongst people is increasing as more people go online to access health information. Health is a widely researched topic on the internet. People research for health and wellness more on the internet compared to seeking direct professional help [96]. Leading online health information tools are general search engines (67%) and health portals (46%)[96]. With the advent of the social web, the next generation web technologies and applications for health(Health 2.0) are emerging as a strong segment with 34% of consumers using social resources such as blogs and forums to locate health information. [96] Tools include Yahoo Answers-style question-and-answer sites (such as WebMD forums) and Wikipedia-style community-authored knowledge (such as OrganizedWisdom).

Health information can quickly get outdated with new diagnosis techniques, treatments and remedies. Acceptability and appropriateness of a source of information becomes vital when you are addressing a personal area like 'Health'. The natural solution when it comes to health is to 'talk' about it. Talking is a natural solution because it is easier to talk about it than formulate an appropriate keyword query for search. Also, one may want to converse like talking to a doctor and explain the situation. Community question answering sites have been, in fact, proliferating on the web in recent times due to this natural method of information access and interaction.

Cobot provides a platform that brings relevant information and connects users through real-time conversations about their domain specific issues. An intelligent agent ("conversational bot") monitors the conversation and provides relevant real-time recommendations. An intelligent agent ("community bot") monitors the community and connects people with relevant conversations and other users. A system like Cobot with socio-semantic information consumption, knowledge processing and filtering has several vertical applications in healthcare decision support, social learning and Information Technology enabled services.

1.3 Search Technologies: An Evolution

Search engine technologies are a practical application of information retrieval (IR) to large-scale document collections. With significant advances in computers and communications technologies, people today have interactive access to enormous amounts of user-generated content on the Web. This has spurred rapid growth in search engine technology, where search engines are trying to discover different kinds of entities such as users, messages, answers to questions or other precise information nuggets found on the Web with emphasis on real time information access.

Semantic approaches to IR use knowledge-based techniques of retrieval that broadly

rely on the syntactic, lexical, sentential and discourse-based levels of knowledge understanding. Semantic approaches include different levels of analysis, such as morphological, syntactic, and semantic analysis, to model, extract and reason from information sources more effectively. The development of a sophisticated semantic system requires complex knowledge bases of semantic information as well as retrieval heuristics. There are a few natural language search engines such as Hakia[1] and Powerset [2] (now part of Bing) that aim to understand the structure and meaning of queries written in natural language text, generally as a question or narrative.

Agent-based approaches [28] involve the development of sophisticated artificial intelligence systems that can act autonomously or semi-autonomously on behalf of a particular user, discover and process information, e.g. [7]. Intelligent Web based software agents search for relevant information using characteristics of a particular domain to organize and interpret discovered information. Personalized Web agents are another type of Web agents that utilize the personal preferences of users to organize search results, or to discover information and documents that could be of value for a particular user. User preferences could be learned from previous user choices, or from other individuals who are considered to have similar preferences to the user. Cobot system is being built as a socio-semantic agent that analyzes information and user activities to provide user specific semantically analyzed information.

The traditional view of Web navigation and browsing assumes that a single user is searching for information. This view contrasts with previous research by library scientists who studied users' information seeking habits. Recent research has demonstrated that additional individuals may be valuable information resources during information search by a single user. Studies have shown that there is often direct user cooperation during Web-based information search. Some studies report that

[1] www.hakia.com
[2] www.powerset.com

significant segments of the user population are engaged in explicit collaboration on joint search tasks on the Web. Active collaboration by multiple parties also occur in certain cases; at other times, and perhaps for a majority of searches, users often interact with others remotely, asynchronously,and even involuntarily and implicitly. Socially enabled online information search (social search) is a new phenomenon facilitated by recent Web technologies. [58] Collaborative social search involves different ways for active involvement in search related activities such as co-located search, remote collaboration on search tasks, use of social network for search, use of expertise networks, involving social data mining or collective intelligence to improve the search process and even social interactions to facilitate information seeking and sense making. Social psychologists have experimentally validated that the act of social discussions has facilitated cognitive performance[119]. People in social groups can provide solutions (answers to questions), pointers to databases or to other people [39][49] (meta-knowledge),validation and legitimization of ideas[46],and can serve as memory aids [63] and help with problem reformulation. Guided participation is a process in which people co-construct knowledge in concert with peers in their community. Information seeking is mostly a solitary activity on the Web today.

The problem we are addressing differs from traditional search paradigms in some ways. Our focus is conversation centric information access; it is not acceptable to return hundreds of results matching a few keywords even if two or three of the top ten are relevant. Unlike traditional information retrieval, the problem requires synthesis of information; it is not acceptable to return a laundry list of results for the user to wade through individually but instead the system must analyze the results collectively and create a solution for the user to consider. And unlike traditional database search, the users are both experts who know how to ask the appropriate questions and non-experts who have more difficulty in knowing the exact question to ask or the exact database query to pose.

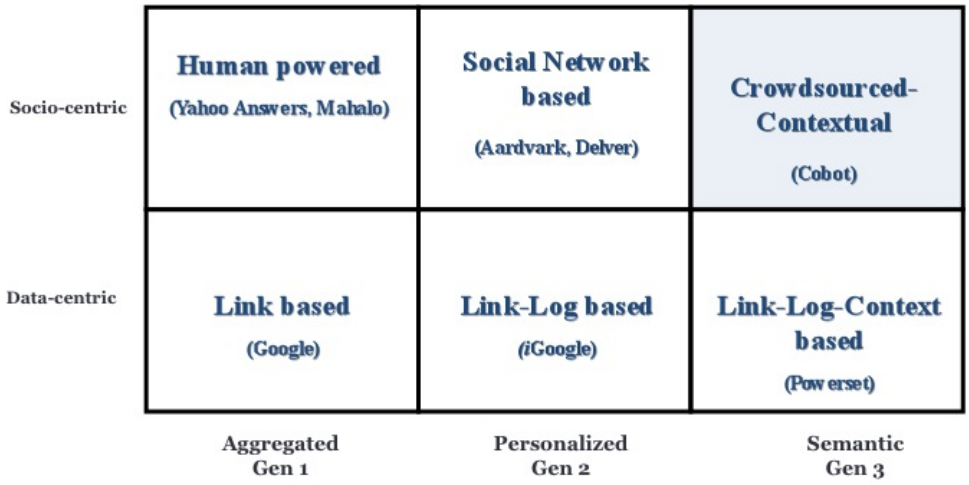

Figure 1: Online Information Access Technologies

Figure 1 captures online information access technology space. We've divided it into data-centric and socio-centric on one axis and aggregated, personalized and semantic on the other axis. Cobot falls in the socio-semantic space. It is social because it uses the user's implicit bonding capital network to find relevant recommendations. It is semantic because it uses knowledge to analyze the conversation and generate meaningful queries. It is augmented because it enhances the conversational experience with integrated recommendations and feedback.

The medium of online conversation allows for sharing ideas, asking questions or discussing issues and solutions interactively along with others. It is an age-old communications practice that helps cultivate creativity, exploratory ideas, perspectives and experiences to take better decisions individually or collectively in the process. Several problems persist with using existing search tools as a means of learning, investigating or exploring about some complex and open-ended information topic. Collaborative social search involves different ways for active involvement in search related activities such as use of a social network for search, use of expertise networks, involving social data mining or crowdsourcing to improve the search process.

The goal, we envision, is to move search from being a solitary activity to being a more participatory activity for the user using natural dialog conversations mixing

social search with traditional web search techniques. In fact, with the vastness of information that exists today, search technologies, as we know it, are blending seamlessly with question answering and recommendation technologies. [1] provides a good overview and directions for web search. In cobot system, the agent performs multiple tasks on behalf of participants of the conversation, and brings in relevant information besides connecting users together. This framework is different from classical IR or Question Answering (QA). The focus of classic IR systems is on retrieving relevant documents from a large document collection in response to a query. While QA deals with more complex understanding of natural language queries, it does not involve a back and forth interaction to continuously monitor, adapt and explore a continuum about some information or questions. This Conversational approach helps users search, explore and ask questions in natural language, leaving the task of user intent comprehension on the system, while the conversational search agents bring together people and different artifacts like documents and conversations together to provide a knowledge-rich participatory atmosphere. Cobot uses technology for operationalizing a user's intent into computational form, dispatching to multiple, heterogeneous services, gathering and integrating results, finding people in the community who best match the ability to respond to user's request and presenting them to the user as a set of recommendations for this request. This conversational framework process involves a series of dialog interactions, agent recommendations and feedback activities.

1.4 The Problem

The length of search engine keywords continue to grow as people now seem to be addressing longer informational and transactional needs as web searches. As opposed to queries that only contained a single keyword, searchers are now using three and four phrases as the standard. Eight or more keywords in the search queries have also grown by 20% as compared to last year (2008)[57]. However, there are very few web

search engines that effectively handle long natural language search queries. In fact, Google, today, limits its search queries to a maximum of 32 words.

In a recent research survey [65] studying how automobile shoppers interact with search engines, a research consulting firm Kelton Research reported some interesting findings. Nearly 40% of Americans (of 1001 people surveyed) describe finding the right and relevant car-related information on the big search engines such as Google and Yahoo overwhelming and time-consuming, according to the survey. They report problems like information deluge, disorganized results and inability to understand keywords. They also reported that 65% population have spent two hours or more searching for specific information on search engines in a single sitting.

Also, in community question answering systems such as Yahoo Answers, users post a question and someone else responds, generally in a very short time, but there is no provision for social search to be in-built in the system. We believe that such a social recommendation feature, with right community balance and the long tail of readily available human workforce, will add immense value to the community question answering system.

Another significant problem in social search is the problem of modeling changing user preferences effectively. The intentions for browsing and searching differ in discrete searches and many social, cognitive, contextual and temporal factors work together in eliciting user preferences. Figuring out the right user models for conversational recommendations is another problem this research is looking at.

The problem we address in this thesis lies at the continuum between web search(laundry-list of results) and question answering (exact nuggets) taking a discovery oriented recommendation based approach that requires domain knowledge for knowledge synthesis. The problem has the following characteristics:

Diverse users: The user may be an expert user like a doctor searching for specific

technical information, a student looking for learning resources, another student looking for pointers to their assignments, a patient searching for disease specific symptoms and treatment options any layman user with a biomedical information need (such as pain management).

Specialized knowledge bases: Knowledge bases are not as large or as diverse as the entire World-Wide Web, yet they are unstructured, contain free text documents, and may not share semantics or ontologies between them.

Relevance: Search queries are longer than average length queries for web search, and they are unlikely to contain all the right keywords (hard to formulate, complex domain, exploratory searches). Yet it is not acceptable to return dozens or hundreds of irrelevant results, even if the right information nuggets are contained amongst them. The aim is to retrieve successive recommendations that try to address the context informatively.

Knowledge synthesis: The user expects the system to provide a relevant pointer enhancing the knowledge about the conversation context and not simply providing a list of documents to read in which the answer may be buried. The system needs to process and correlate information from multiple sources, and form multiple filtering strategies so as to develop a specific recommendation for the user.

Community specific: Users of the community talk about certain kinds of problems and issues thereby creating certain kinds of implicit social bonds in the community. The recommendation system should take the past community interactions into account and the recommendations should get more community-centric with time.

1.4.1 Research Objectives

The specific research objectives addressed in this thesis are:

1. Blended Platform: *How do we use AI/Informatics/Information and Communication Technologies(ICT) techniques to help consumers find relevant information with less effort?*

2. Knowledge based Information Architecture: *How do we bring precise semantic information to the user who is looking for specific problems and situations?*

3. Community focussed: *How to connect to specific people who have the knowledge the asker is looking for?*

4. Evidence based: *How do we build a recommendation engine that processes information from different sources, bringing in new information as well as looking at past information applying different filtering strategies and selecting the best matching recommendations dynamically for the conversation?*

1.5 Solution

The rapidly increasing volume of unstructured textual information poses the challenge of knowledge integration so as to build autonomic computing systems that can acquire, represent and learn such knowledge, and efficiently reason from it to aid in knowledge discovery and re-use. The construction of these automated systems to assist decision making is impeded by difficulties in formalizing knowledge and in encoding that knowledge for use by computer systems. This research focuses on developing knowledge based methods of information retrieval, extraction and recommendations and build a socio-semantic infrastructure using techniques of language processing, modeling and recommendations.

Conversational Information Access is an interactive and collaborative information finding interaction. The participants in the conversation engage in a conversation activity aided with an intelligent agent that provides conversational recommendations.

Figure 2 shows the three different aspects of the agent that this thesis focusses on

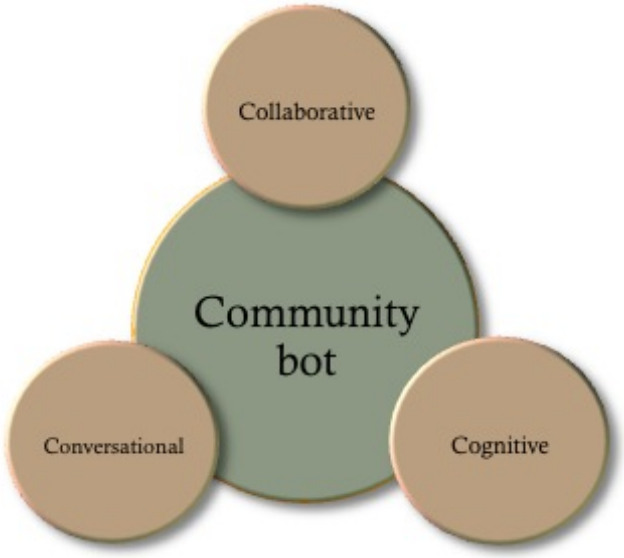

Figure 2: Cobot Features

- conversational recommendation aspect, the collaborative system aspect and the cognitive user and knowledge model aspect. These three different dimensions of the agent makes the system a community based humanly helpful recommendation system.

CIA is an augmented social search and recommendation activity with the goal to interactively engage in conversations and receive agent supported recommendations. It is useful because people engage in online social discussions unlike a solitary search; the agent brings in relevant information as well as identifies relevant users; participants provide feedback to the agent during the conversations that agent uses to improve it's recommendations.

CIA is different from Information Retrieval (IR) or Question Answering (QA). Information Retrieval has focused on retrieving relevant documents and passages from large text corpora. This focus is a perfect match for a variety of tasks such as those found with navigational searches. If the user's information need is more specific, browsing complete documents for answers to questions is time consuming and inefficient. Moreover, IR generally does not deal with the process of understanding the meaning of queries when posed in natural language, e.g. in the form of a question

or paraphrases.

In Question Answering (QA), researchers are developing different algorithms and techniques to obtain effective responses for specific information requests. The solution is generally present in a paragraph, sentence, or phrase. These snippets of information contain possible answers to the posed questions. While QA deals with understanding the meaning of natural language queries, it does not involve a back and forth interaction to find out about some information or questions.

CIA involves exchange of information between the sender and recipients; the agent has to pay attention to the information flow, analyze the question and responses in conversation and provide recommendations to fulfill the conversational information need. It involves techniques involved in both IR and QA.

There are several challenges in CIA besides the inherent problems in IR and QA. Some of the additional problems in CIA are:

- *How to model CIA as a collaborative information finding activity?*
- *How do we apply the model to provide recommendations?*
- *How do we dynamically connect cohorts based on the conversations?*
- *How do we evolve the interaction model to understand the conversation and conversation flow?*
- *How does the agent adapt to user preferences while providing recommendations?*

In this thesis, the approach we have taken to address CIA is by developing a socio-semantic conversational recommendation platform, using knowledge from tags and ontologies and developing user models and implicit social bonding network with semantic filtering to drive the recommendation process.

1.5.1 Socio-Semantic Conversation Model

"The core problem that context-sensitive asynchronous memory addresses is how to get the information an agent needs when it doesn't know how to ask the right question and doesn't have the time to exhaustively search all information available to it. The key to this solution is to interleave remembering with thinking and doing, thus making the context of thought and action available to guide remembering." [50]

The Socio-Semantic Conversation Model that we envision is a dynamic memory data structure based on principles of experience based agent architecture[88]. It supports interleaved retrieval of information by applying different memory retrieval algorithms such as Spreading Activation. The model maintains the user's social graph, the conversation graph with the extracted semantic net for the conversation.

Some essential properties of the model are as follows:

- *The model should provide for a way to register feedback (for learning)*
- *The model should be aware of the participant's interactions (to aid Cohort Matching)*
- *The model should allow for domain knowledge incorporation (for queries and knowledge based user models*

Figure 3 displays this integrated socio-semantic approach combining data and related meta-data (information) with case knowledge and interactions (experiences) to create an integrated socio-semantic model.

1.6 Thesis Organization

Figure 4 captures the problem addressed, and approach taken to address the problem (solution) and the next level sub-problems associated with the broader solution and so on. The tree depicts the problem of information access and solving it using the "Conversational Recommendation" approach.

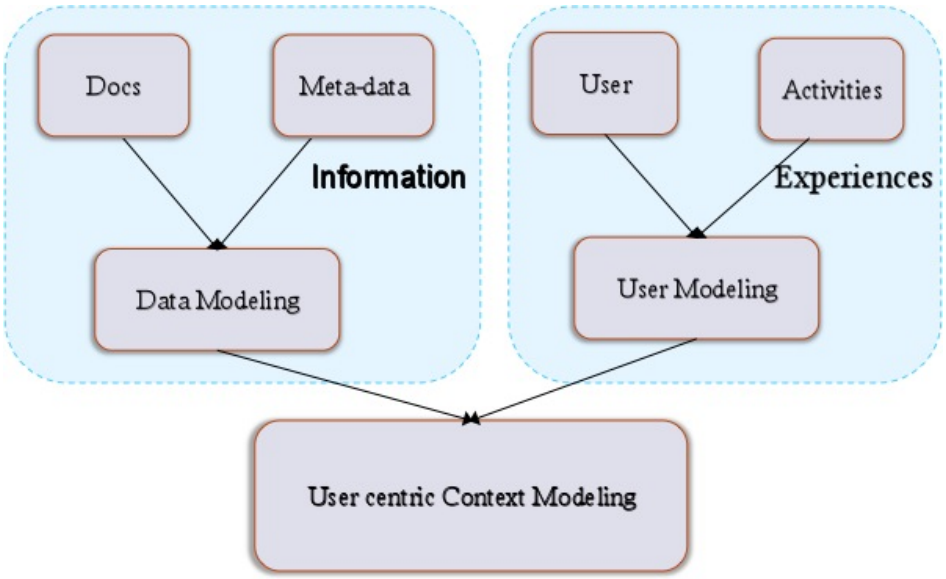

Figure 3: User-centric Domain Information Modeling

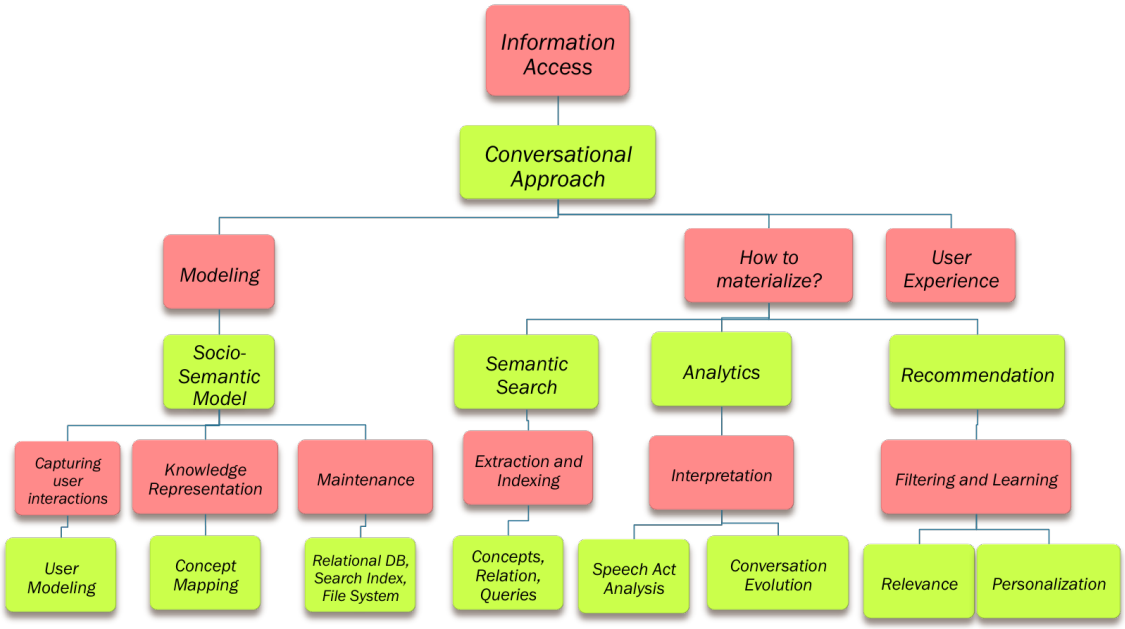

Figure 4: Thesis Tree

- *Socio-Semantic Model* This thesis develops an integrated socio-semantic natural language conversational recommendation platform that provides multi-modal modeling and recommendations of users and documents.

- *Cohort Matchmaking* This thesis describes a methodology to dynamically recommend users for conversations (Cohort Matching). Unlike users themselves having to find relevant conversations, the conversations find the users using this approach.

- *Scope for Application to other domains* This thesis describes generic methods and algorithms for information access applied to the health and some science domains. These methods and algorithms are not domain specific - replacing one rich domain specific knowledge with any other knowledge would port the system to that domain.

CHAPTER II

CONVERSATIONAL RECOMMENDATION COMPONENTS

In this chapter, we will briefly outline the architectural and functional components of a socio-semantic conversational recommendation system. These features and components have been incorporated in cobot system.

The key dimensions of a conversational recommendation system include both relevance and timeliness of recommendations. For complex domains, for example, healthcare, there may be additional key dimensions such as credibility of recommendations that become critical for the success of the system. To construct a successful conversational recommendation experience, it is critical to build an effective socio-cognitive experience keeping these features in consideration.

We briefly elicit the key features of a community based information access system as we've envisioned in Cobot:

1. *Filtering:* Information filtering and a push based recommendation technology are crucial in today's online information access systems. These technologies enable users to navigate and manage an ever-growing deluge of information more effectively. Cobot recommendation engine delivers recommendations processed through filters for identifying concepts, properties of concepts and intentions behind the conversation. It also uses various personalization filters from knowledge based matchmaking to social filtering based on past shared interactions between community users.

2. *Notification:* Users should be dynamically notified of relevant information updates with respect to a users' community and his conversations. Users should

also be able to follow other users and conversations as well as have a personalized feed of information based on their interests and prior conversations.

3. *Multi-modal Recommendations:* Cobot provides different kinds of recommendations that are real time and dynamic. Not only does the system provide article based recommendations, but also it connects relevant people to conversations proactively using a socio-cognitive matching engine.

4. *Collaborative Engagement:* Cobot system has its foundations laid in the principles of user-generated content thus making collaboration aspect key to the system. The system allows for users to leverage each others' conversations and recommendations by rating the recommendations (of all types: people, conversations and online resources) and thus building social trust in the community and its content through engagement.

5. *User Models:* User models are the system representation and understanding of a users' interests and needs. It allows the system to perform the required filtering and provide the user with relevant recommendations. Cobot builds models of users' knowledge and interactions by capturing implicit and explicit feedback and building short term and long term models for the user.

6. *Conversations:* Cobot system is built around the concept of conversations. The differentiator between regular search engines or information portals and a conversational information access system is that CIA brings information to the user without search, implying that it understands the user's needs and pushes the required information to the user without her go out on the web and perform a solitary searches with the added load of trying to figure out right search queries.

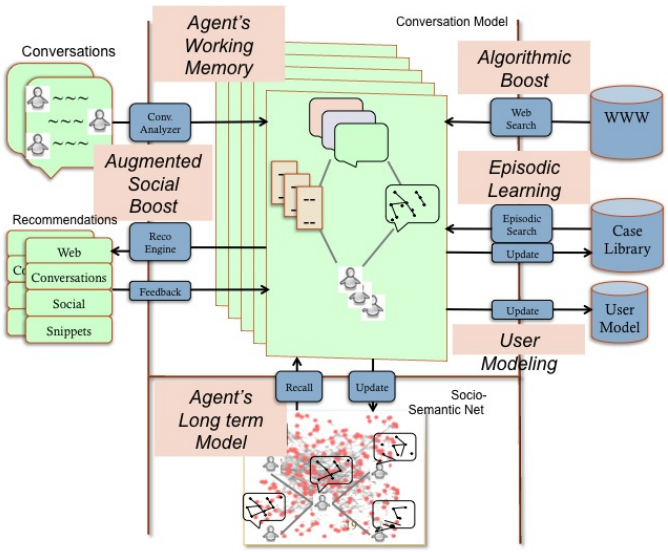

Figure 5: Conceptual Framework

2.1 Architectural Components

Figure 5 shows the conceptual framework and components of cobot architecture and how the agent analyzes conversations, brings in external knowledge, leverages from past interactions, stores and recalls from the knowledgebases and finally provides recommendations to users.

Knowledge explosion continues to outpace technological innovation. It is increasingly difficult to find relevant information, not just on the World Wide Web at large but even in domain-specific medium-sized knowledge bases (such as sites like WebMD, PubMED, or CDC.gov for healthcare). Search results are not tailored to the users' goals or information need, or to his/her specific medical situation. There are further technical challenges in biomedical domain community information system because of the complex medical language and terminologies, varied outcomes and different individual experiences on similar situations.

Cobot system is agent based and agent assisted. We browse, find and soon forget what we have found. The agent based system maintains and uses case based reasoning to help the users quickly find relevant information from the web or his past interactions

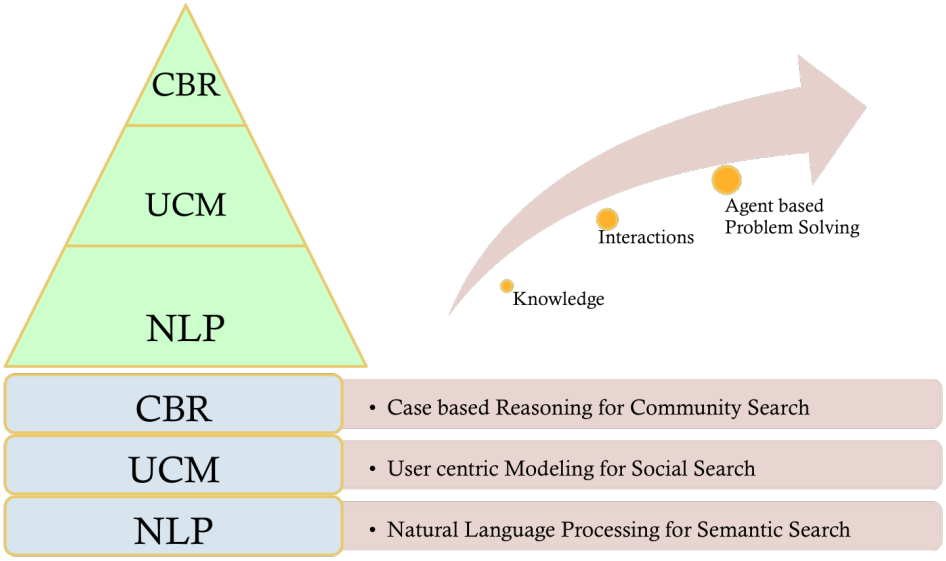

Figure 6: Key Ideas

and experiences. Figure 6 shows the key ideas that we've built upon in cobot system and architecture.

In the following sections, we briefly outline the architectural components of cobot system.

2.1.1 Precise Search

Identifying relevant documents for a particular user's need without extensive search, in conversational manner is the key objective here. Search queries may be much longer than five to seven word standard queries typical for web search, they may be unlikely to contain all the right keywords. It is not desirable to return dozens or hundreds of remotely relevant results, even if the right answer is amongst them. The aim is to retrieve successive solutions as an interactive experience that try to address the access problem precisely.

2.1.2 Knowledge Synthesis

The user expects the system to provide facts and experiences and not simply a list of documents to read in which the answer may be buried. The system needs to integrate and correlate information from multiple documents, from multiple data sources, and/or from multiple reasoning strategies so as to develop a specific recommendation for the user. Also, recent advances in mining the social web have led to work towards modeling users and social interactions across the web. Combining the document models with the user models in an integrated representation will lead to development of systems that intrinsically lend its model to user centric personalization efforts. We are developing a graph based representation of our information model that includes data entities as well as user based entities.

2.1.2.1 Semantic Model

A semantic network [104] is a structure of interconnected nodes and links for representing knowledge. The nodes can represent terms, concepts, classes and objects and the links can represent relations between the nodes such as roles and properties. The generic notion of semantic networks has been extended to work on some very powerful connectionist knowledge representation systems using logic and deep theoretical foundations. These approaches have laid the foundation for the current semantic web infrastructure.

Collins and Quinlann [36] studied ways to capture words and meaning from natural language and represented them as the earliest semantic networks. The connections within the above model were not only associative in nature, the links between nodes were qualitative and purposeful as well. Another parallel work from the field of linguistics gave rise to grammar and tree based approaches [30] [31] that resulted in deep natural language processing foundations. Schank and colleagues [98] [99] developed conceptual dependency approach where the attempt was to break down the sentences

into a network of meta-concepts that were universal and language independent.

As an extension to these broadly defined semantic networks, a lot of related work has been done in the field of ontologies. Ontologies are an explicit formal specification representing objects, concepts, and the relationships among them within a defined area of interest. They are usually hierarchical and interconnected. Ontologies provide a standardized vocabulary for representing and communicating knowledge about objects and their relationships to one another. Because the ontological terms and the relationships between them are carefully defined by domain experts, the use of ontologies helps standardize annotations, improve information retrieval, and supports the construction of inference statements. It is generally believed by the scientific community that ontologies can make a significant contribution to the design and implementation of better and more interoperable information systems [85].

There is an ever-increasing need for a strong conceptual foundation for data sharing to give precise semantics to the heterogeneous data existing in different repositories. Ontologies not only make knowledge re-use easier, they are also the foundation standardization efforts since they make explicit the conceptualizations behind a terminology or a model. [84]. Automatic knowledge acquisition into an ontological paradigm, where data can evolve and be shared, thus provides a formal framework to this information management process.

2.1.3 Case based Reasoning

Case-based reasoning is an artificial intelligence approach, in which past cases are used to solve new problems [67] [73]. The key lies not in running a smarter search engine against a set of documents, but in understanding which documents contain appropriate answers to users' different kinds of queries using their past experiences. While driven by information retrieval techniques, there is a learning component that goes beyond simply matching queries against documents to matching queries against

past episodes. Cases are stored in a case library and represent the acquired experience or historical record of previous queries and responses.

A typical case-based reasoning (CBR) system works as follows. Given a new problem:

- retrieve a few past cases from a case library that are close to the new problem in a suitable representation space;
- for each retrieved case, calculate a similarity metric between the case and the new problem and select the best match;
- apply the solution in the selected case to the new problem;
- learn by modifying the proposed solution based on feedback from execution and storing it back in the case library.

Our work uses the familiar CBR cycle (retrieve, select, apply, learn) but with the following differences:

- there is a separate acquisition and representation phase which builds the knowledgebase.
- retrieve and select require text analytics (in our case, NLP, search), since the knowledgebases and cases are unstructured text instead of traditional AI representations
- learn requires human-in-the-loop relevance feedback and requires storing new cases in the case library

Instead of matching queries against keywords in documents, the system develops a case library of past problem-solving sessions containing previous queries the system has seen and corresponding solutions the system has proposed. The key research

question, then, is how a system can perform case-based reasoning with textual information; how it indexes, retrieves, selects, synthesizes and how it learns by interacting with the user. More specifically, this approach raises the following research issues:

- Knowledge Representation: What information does a case contain apart from the given knowledgebase representation? How is this information represented? How is the case library initialized and what happens if there are no past cases for a given query?

- Indexing and Retrieval: How are cases organized to enable relevant cases to be found later? How are cases retrieved in response to a user's query? How is the relevance of a case determined?

- Decision Making: How are multiple cases combined to produce the final answer(s)? How is the level of confidence in an answer determined?

- Learning: How are new cases learned? How are indexes and cases updated through experience?

Case based Reasoning is applied to knowledge as well as myriads of user experiences over the web. We have proposed 'Discover, Represent, Interpret, Learn' phases as depicted in Figure 7 for the web information experience reasoning architecture.

Effective problem-solving necessitates a situation assessment phase. In this discovery phase the system needs to maximize its understanding of the problem. The query is converted from the language of communication to the system understandable language of representation, depending on the choice of representation structure selected by the user. In the context of textual CBR, the system can improve its understanding of the problem in two ways - 1) by knowing more about different concepts related to the language of communication 2) by knowing more about the concepts related to the query terms in the language of representation. Linguistically, knowing more about

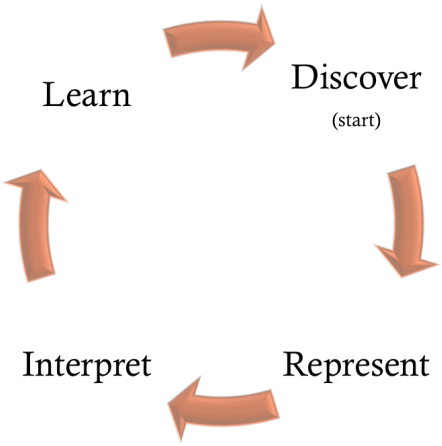

Figure 7: CBR Phases

the query-related terms helps the system to understand and interpret the different ways in which the query terms may be represented in the representation language. The learning phase involves revising and storing the problem solution cases.

2.2 Functional Components

In the rest of the chapter, we describe the functional components of cobot system that help materialize the architectural components.

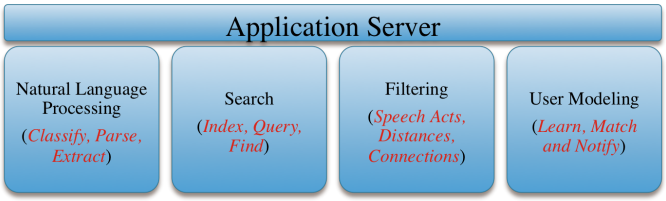

Figure 8: Semantic Components

Figure 8 shows the backend cobot engine running as a web service with functions for NLP, Search, Filtering and User Modeling.

The main functional backend components of a conversational information access system can be classified into the following functions:

1. *Language Understanding*

2. *User Modeling*

3. *Filtering and Recommendations*

We briefly describe each component for conversational information access in Cobot system.

2.2.1 Language Understanding

Figure 9 shows the Semantic Analyzer that extracts concepts, utterances and relevant queries from conversations.

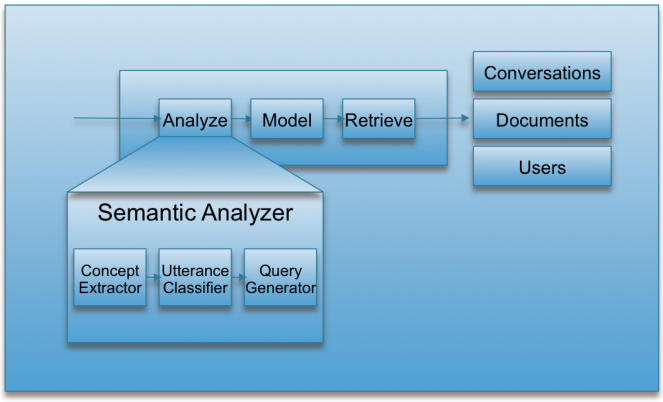

Figure 9: Semantic Analyzer

2.2.1.1 Intent Detection

Conversational interactions are classified into one of the following categories in Cobot to strategize for query reformulation stage and to help make the decision if the agent should insert some type of recommendation into the conversation:

- *Question*: Asking a question, e.g. somebody posts a problem. This is usually, but not always, the first post of a thread.
- *Disclosure*: Reveals thoughts, feelings, wishes, perceptions or intentions (declarative first person)
- *Edification*: States objective information

- *Advisement*: Attempts to guide behavior - suggestions, commands, permission, prohibition
- *Acknowledgement*: Being recognized or acknowledged
- *Reflection*: Repetitions, restatements and clarifications
- *Interpretation*: Judgement or evaluation of other's experience or behavior
- *Confirmation*: Compares speaker's experience with other's agreement, disagreement, shared experience or belief

2.2.1.2 Query Generation

Cobot analyzes conversations to extract concepts, relationships between concepts and focus of conversations to generate meaningful queries for external search engines for bringing in relevant candidate results. We use OpenNLP chunker trained on medical corpus [43] to extract phrases and map them into concepts using UMLS ontology [11]. Main concepts expanded with their synonyms in conversations help us in retrieving recall oriented documents. We extract SVO triples [93] from sentences as queries to retrieve documents that closely match the context in conversations. We also experimented with generation of queries based on the predicate argument structure in sentences using ASSERT semantic role labeling system [87] but removed it out of our deployed system due to increased processing times and our near real time access requirement for the system.

In cobot, we have built a fast shallow semantic parser (Figure 10) capable of extracting relationships, phrase focus and their properties using Augmented Transition Networks.

Figure 11 shows an example query candidate extracted from our parser.

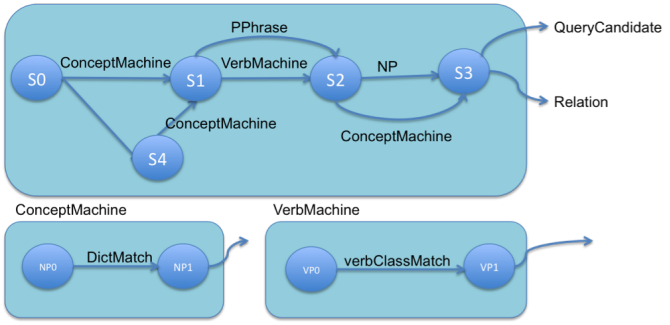

Figure 10: ATN Parser

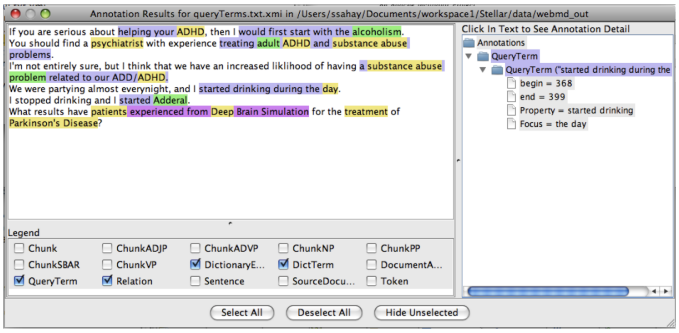

Figure 11: Example - Query Candidates

2.2.1.3 Semantic Tagging

Socially enabled systems have the property of self-governance and evolution by its community. While the major challenge remains getting a critical mass, these require lesser coordination. The problem with social tagging is that the noise-signal ratio becomes high due to informal nature of the language in conversations. Cobot system normalizes these conversations to extract meaningful conceptual representations using the extensive UMLS ontology and fast approximate matching to guide social tagging of conversations. Cobot's internal knowledge representation system uses the concepts from UMLS and Wordnet as it's language of representation. Figure 12 shows an example concept extraction from a document in cobot.

2.2.2 User Modeling

Language and interaction (percepts) creates usable memories, useful for making decisions about what actions to take and what information to retain. Cobot leverages

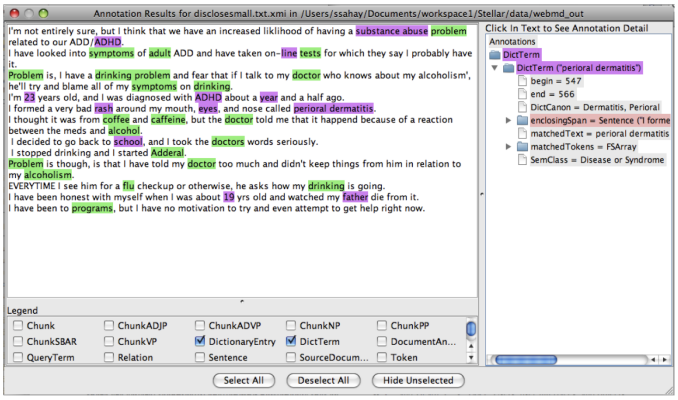

Figure 12: Example - Concept Extraction

these interactions to maintain users' episodic and long term semantic models, agent's per conversation working memory of concepts, syntactic and semantic information nuggets, and participating users and messages. The agent analyzes these memory structures to bring in external recommendations into the system by matching with the contextual information need. The social feedback on the recommendations are registered in the indices for the algorithms to generate their user specific and conversation specific contextual relevance.

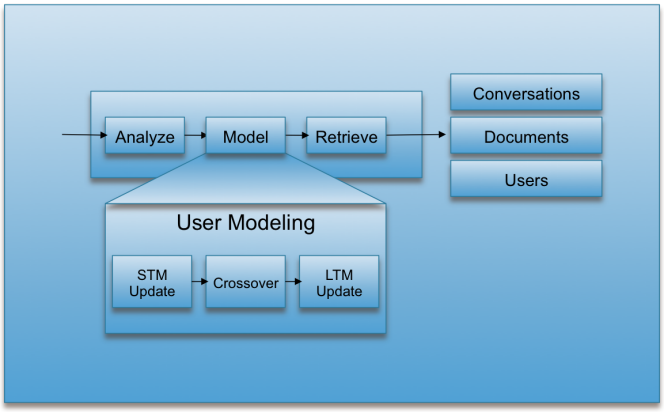

Figure 13: User Modeling

Figure 13 depicts the user modeling pipeline in cobot. The purpose of Episodic Memory is to capture the user's short-term interactions and interests. Based on user's frequency of interactions and diversity in topics, this memory empirically varies in the range of a few days for different users. The Semantic Memory captures the

user's long-term profile. These are the topics that interest the user in general and for a prolonged time. These interests change less frequently and represent general criteria of recommendation to the user. Many times, users might be interested in some temporary information need. Such information need not be incorporated in the long term user memory. The episodic memory captures such short-term interests. The episodic memory forms a sort of staging area and the concepts from this memory are selectively and periodically moved to the semantic memory in a crossover process.

The nodes of the semantic memory are concepts extracted from user's interactions. The concepts are connected with associations which develop when concepts co-occur frequently. Over a period of time when the user participates in more interactions, new concepts are added to the semantic memory. Our system currently tries to find a recently active user first who participated in similar conversations. Different conversational facets are matched with episodic memories and a spreading activation search on the semantic net is performed for recommending the top 3 users for the conversation. The activation is spread to the neighboring nodes proportional to the weight of each connecting association in the semantic net. There are several parameters in the system that can be learnt based on activity of users. Parameters for episodic memory window size, semantic memory learning and unlearning rates, concept co-occurences and feedback strengths for associations are initially set heuristically and can be fine-tuned to suit individual users.

2.2.3 Recommendations

Cobot provides three types of recommendations. It recommends and notifies relevant people who may be interested in joining conversations. It provides topic specific web recommendations and it also suggests past similar conversations from the system. Figure 14 depicts the Retrieval Engine that fetches conversational recommendations.

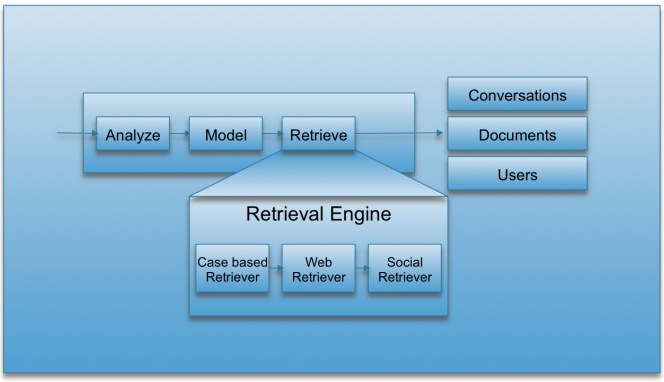

Figure 14: Retrieval Engines

2.2.3.1 People Recommendation:

While designing a recommender system, it is important to take into account the domain implications and fine-tune the algorithms accordingly. To provide social recommendations with a high degree of conversion rate, the system needs to identify people who can provide answers to asked questions, share similar health experiences and provide topic specific opinions and advice. Our system is built around health and education information domain therefore users are generally not concerned with building their social ties, instead, the goal is to serve the user's contextual information need. One important aspect in this domain is reputation of the recommended users, since there is no prior information and relationship of these users with the person who starts a conversation. We have built the reputation system by allowing users with the ability to rate conversations, users and documents.

Different conversational facets are matched with episodic memories and a spreading activation search on the semantic net is performed for recommending the top few users for the conversation. The activation is spread to the neighboring nodes proportional to the weight of each connecting association in the semantic net. There are several parameters in the system that can be learnt based on activity of users. Parameters for episodic memory window size, semantic memory learning and unlearning rates, concept co-occurences and feedback strengths for associations are initially set

heuristically and can be fine-tuned to suit individual users.

2.2.3.2 Knowledge Recommendation:

For web search and conversation recommendations, we reformulate queries from the conversation snippets based on occurrence of concepts and relationships and types of messages. For a given target query Q_t, past community conversations are ranked so that the results which are most likely related to the learned preferences of the community are promoted[103][82][76]. This kind of personalization is based on the reuse of previous search episodes: the promotions for Q_t are those results that have been previously selected by community members for queries that are similar to Q_t. Cobot creates different user communities based on the type of forum users participate in. For example, users in 'Health Sciences' group become part of the Health community in Cobot whereas users in Mathematics group become part of the Mathematics community.

Cases are represented as tuples made up of the query component (a set of query terms, Q_i used during some previous search session) along with web recommendations and past conversations with their community hit counts. Our formulation is based on similar work reported in [103]. Each case is a summary of the communitys' search experience relative to a given query.

Each new target problem (corresponding to a new query Q_t) is used to identify a set of similar cases in the case base by using a term-overlap similarity metric to select the n most similar search cases for Q_t .

These search cases contain a range of different result pages and their selection frequencies. Bearing in mind that some results may recur in multiple cases, the next step is to rank order these results according to their relevance for Q_t. Each result R_j can be scored by its relevance with respect to its corresponding search case, C_i by computing the proportion of times that R_j was selected for this case's query Q_i.

During the development of retrieval stage of the CBR system for Cobot, it was often observed that number of results retrieved were very large since the retrieval stage entailed a meta-search which queried many search engines which returned large number of results. We wanted to show only the top 2 to 3 results /conversations from the retrieved case base. Consequently sorting and ranking results according to their relevance to the ongoing conversation was necessary.

Relevance of a result with respect to the current target query Q_t) is calculated by computing the weighted sum of the individual case relevance scores, weighting each by the similarity between Q_t and each Q_i. In this way, results which come from retrieved cases (C_1, ..., C_n) whose query is very similar to the target query are given more weight than those who come from less similar queries. The relevance of a Result R_j to a target query Q_t and the case library comprising of cases from C_1 to C_n cases is expressed as:

$$WRel(R_j, Q_t, C_1...C_n) = \frac{\sum_i Relevance(R_j, C_i) * Similarity(Q_t, C_i)}{\sum_i Exists(R_j, C_i) * Similarity(Q_t, C_i)}$$

Similarity between the query and case is computed by finding the similarity between the query and case queries. We are using Jaccard Similarity as the similarity metric in our design. In this way, for given user, with query Q_t we produce a ranked list of results R_j that come from the community's case base and that, as such, reflects the past selection patterns of this community. If the case base doesn't retrieve cases or the similarity confidence of the retrieved results is less than a user specified threshold t then, Q_t is used by the meta-search module to retrieve a set of web search results.

The top few results from the ranked results obtained either from the case base or the meta search engines are shown to the user. In this way, results that have been previously preferred by community members are either promoted or marked as relevant to provide community members with more immediate access to results

that are likely to be relevant to their particular needs. This framework promotes community preferred results and conversations to the user.

CHAPTER III

DESIGN AND ARCHITECTURE

Cobot is designed and developed using a combination of Web 2.0 and Artificial Intelligence technologies. Instead of relying on search engines that inundate the user with a multitude of information, Cobot models the information finding task as an interactive user experience within a social community. The user describes her need in natural language to a trusted community (e.g., you might come home and ask a friend whose father had suffered a similar condition). This is modeled via text conversations, which is familiar to most users. Our intelligent Web 2.0 framework uses 'wisdom of the crowds' philosophy mixed with automated conversational recommendations. Such an approach enables the system to make highly personalized recommendations that are tailored to a specific user discussing a specific problem in a specific scenario.

The following design goals differentiate Cobot from other recommendation systems:

Design Goals

1. Mixed Initiative (human centric, agent assisted)
2. Proactive Social (connecting people to conversations)
3. Semantic (Natural Language Knowledge Extraction)
4. Near real time (Instant Notifications)
5. Community based feedback and learning (Agent learns with interaction)

3.1 Workflow

Figures 15 16 17 18 depicts the experience through the system.

Figure 15: User starts a conversation

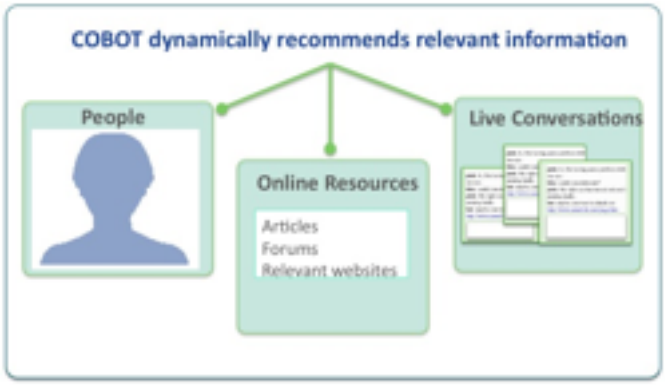

Figure 16: Cobot recommends

Figure 17: User browses and rates the recommendations

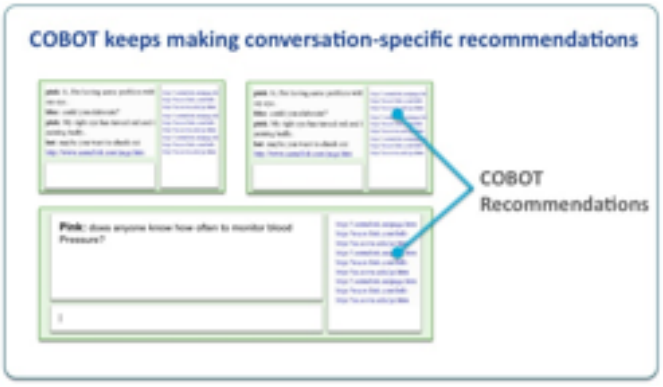

Figure 18: Cobot interleaves more recommendations

3.2 Architecture

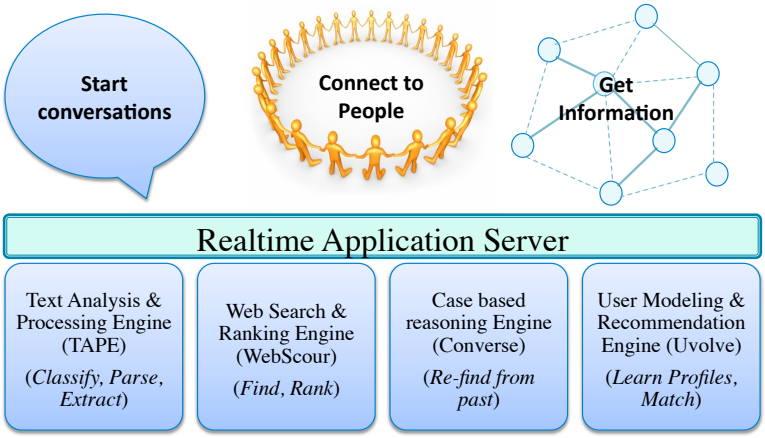

Figure 19: Architecture

Figure 19 describes the Cobot architecture diagram to depict the backend processing involved while a user is actively engaged in a conversation. The conversation agent processes the conversation text and decides whether it needs to generate a recommendation for that conversation. These recommendations come in the form of other users to participate in the conversation, web search re-ranked results and other similar conversations from Cobot's semantic search indices after going through several filters. Cobot system architecture, as depicted in Figure 19, is organized around the following elements:

1. *Dictionaries and Ontologies* Cobot bootstraps on the knowledge provided to it

through domain specific dictionaries and ontologies (constructed from UMLS ontology and StackOverflow tags for Math and CS domain) for processing.

2. *Data sources* Various data connectors are available in Cobot to connect to databases, search indices, large xml dictionaries through configuration files

3. *Communication Infrastructure* Cobot uses Jabber protocol and infrastructure for instant messaging and notifications

4. *Information Processing Framework* Cobot's processor and memory intensive information processing components are built on top of Unstructured Information Management Architecture (UIMA) infrastructure.

5. *Real time Search* Cobot uses Zoie[1] infrastructure as it's search component for real time indexing and retrieval of candidates.

6. *Web Server Infrastructure* Cobot is packaged as a web server technology on top of an Application server for serving clients

7. *Core Tools and Algorithms* The backend engines in Cobot are responsible for conversational understanding and analytics, user modeling and recommendation generation.

8. *Helper Libraries* Cobot uses various helper tools and libraries (example, Weka for classification of speech acts)

Figure 20 shows the Sequence diagram of Cobot for user authentication, inbound requests and asynchronous recommendation responses. The current design easily allows for Cobot to being used as a Web Service in future.

[1]http://code.google.com/p/zoie/

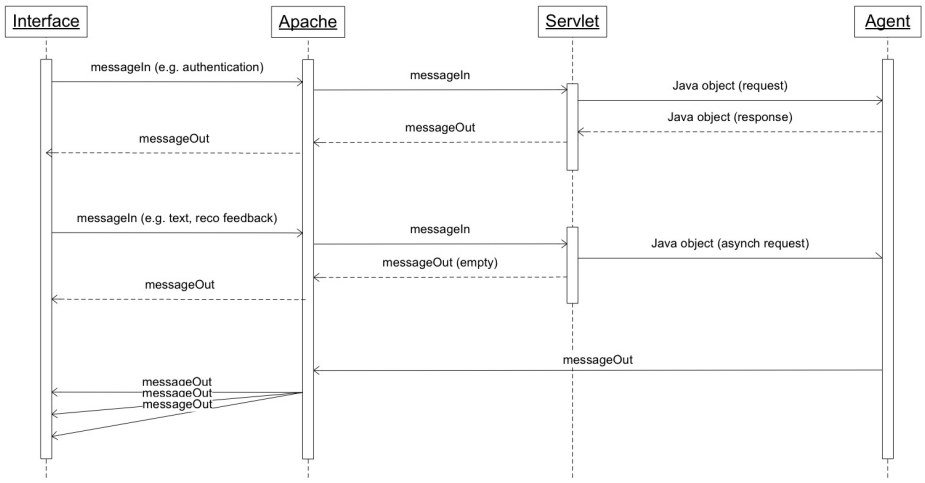

Figure 20: Sequence Diagram

Figure 21 shows the high level packages in Cobot system. The core modules of the system include nlp, search, filters, similarity, profile and annotators. The main cobot entities are shown in the database schema diagram in Figure 22.

3.2.1 Unstructured Information Management

UIMA stands for Unstructured Information Management Architecture, which is an Apache project information processing software framework that provides scalable infrastructure for analyzing large volumes of unstructured information in order to discover knowledge that is relevant to an end user. An example UIM application might ingest plain text and identify entities, such as persons, places, organizations; or relations, such as works-for or located-at. UIMA enables applications to be decomposed into components, for example "language identification", "language specific segmentation", "sentence boundary detection", "entity detection (person/place names etc.)". Each component implements interfaces defined by the framework and provides self-describing metadata via XML descriptor files. The framework manages these components and the data flow between them. Components are written in Java or C++; the data that flows between components is designed for efficient mapping between these languages. UIMA additionally provides capabilities to wrap components as network

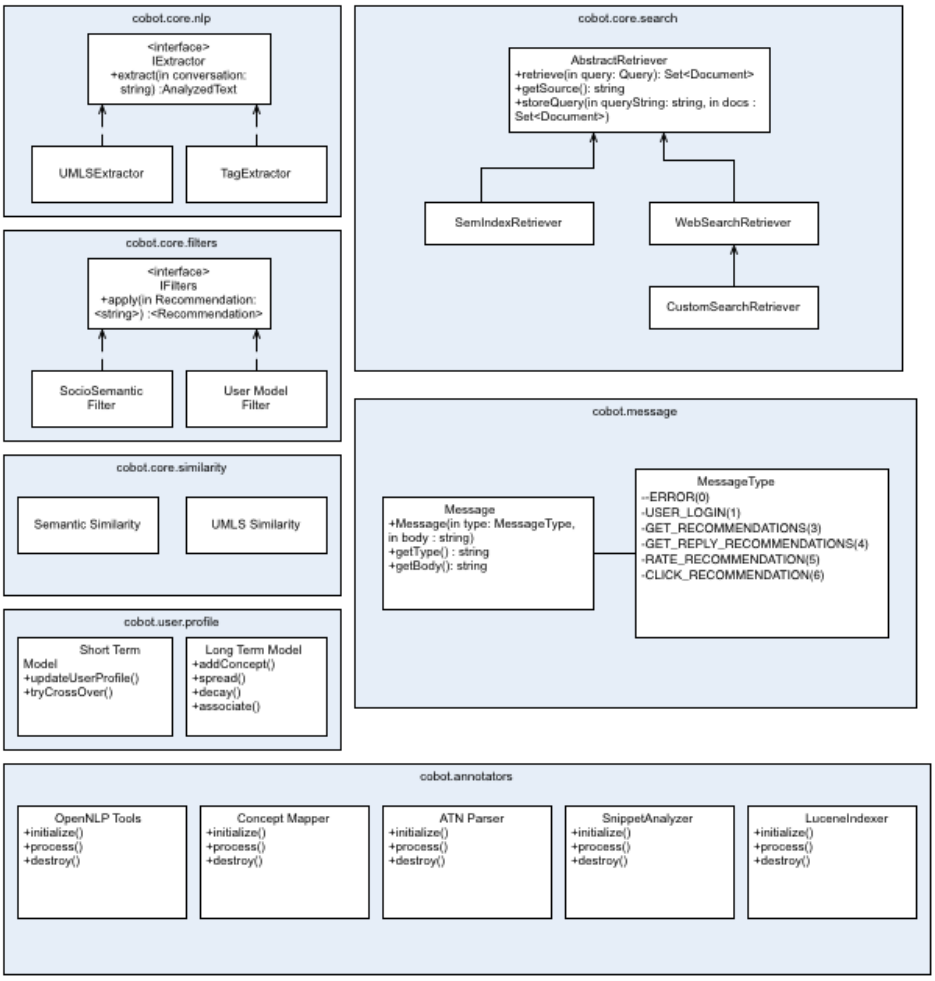

Figure 21: High level classes

services, and can scale to very large volumes by replicating processing pipelines over a cluster of networked nodes.

Figure 23 shows architecture of UIMA framework. As shown, unstructured information including text, chat is input into the UIMA pipeline through a collection reader. Once the information resides in the Collection reader, it is converted into a CAS or a Common Analysis Structure which is a data structure on which rest of the components of UIMA analysis engine run. The Aggregate analysis engine specifically runs various analysis engines on the information residing in a CAS like entity annotations, relationship annotations etc. The CAS produced by the Analysis engines is then either populated in a knowledge base, a database or an index through a CAS

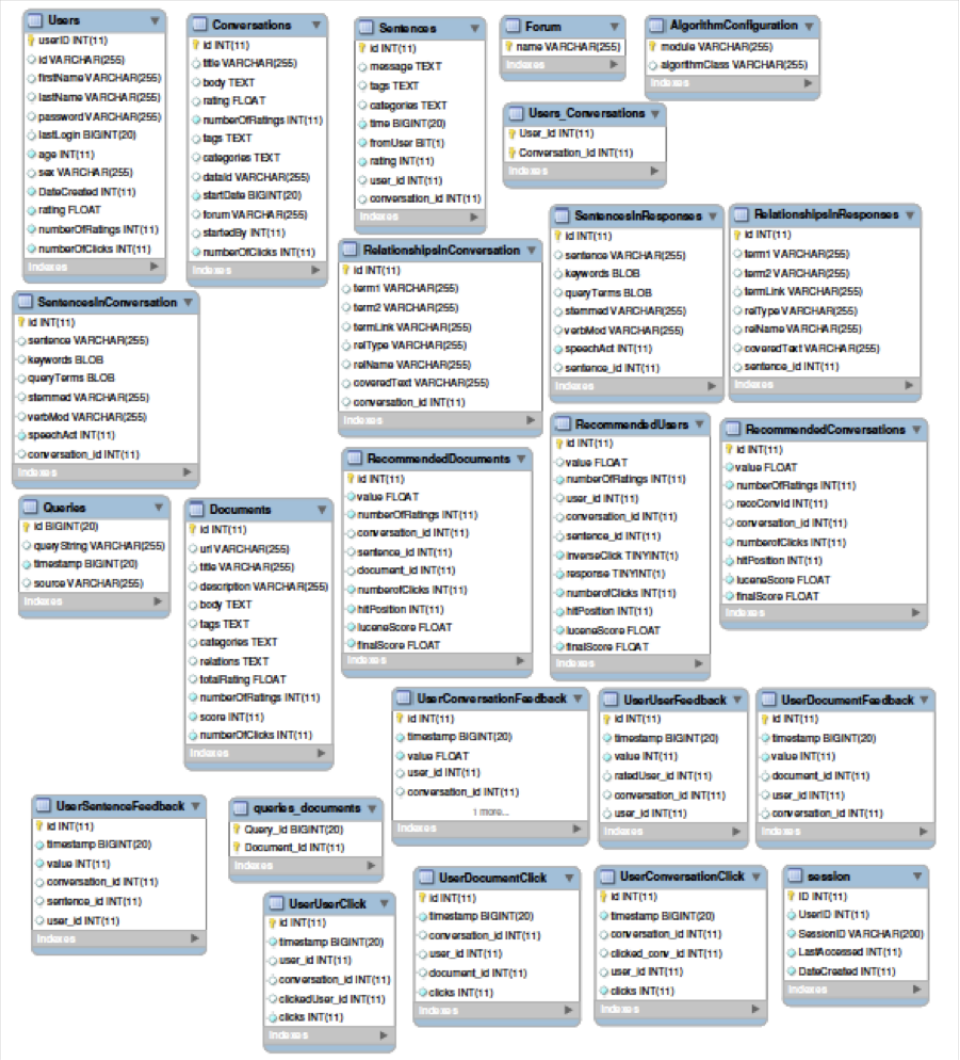

Figure 22: Database Schema

consumer.

3.2.2 Real time indexing and retrieval support

We have adapted a configurable open-source UIMA Common Analysis Structures (CAS) to Lucene document generation system [2] for real time indexing and retrieval in Cobot. We have Conversations, Responses, Webpages and Users in the Cobot database as main first class indexible entities. We map the primary keys of the

[2]http://bit.ly/rbm4rG

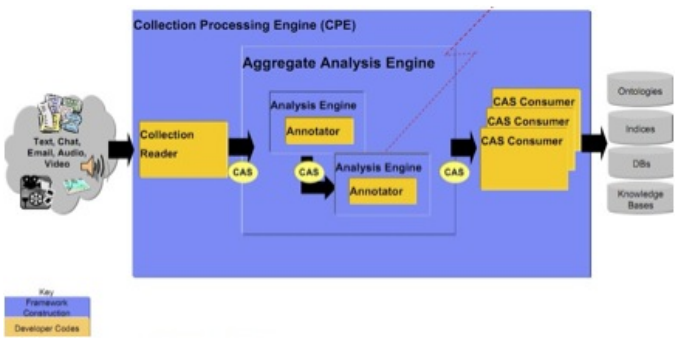

Figure 23: UIMA Common Analysis System (source: UIMA documentation)

Database tables in the indexes for mapping of retrieved results to the database entities. The analysis engines extracts these keys and maps them to unique ID mapper in our real time indexing infrastructure for fast lookups.

We have incorporated Zoie infrastructure for enabling real time support for Cobot conversational access pipeline. Zoie is a real-time search and indexing system built on Apache Lucene. Zoie was donated by LinkedIn.com and has been deployed in a real-time large-scale consumer website. In a real-time search/indexing system, a document is made available as soon as it is added to the index. This functionality is especially important to time-sensitive information such as news, job openings, tweets etc. This has not only resulted in replacement of non-scaling database based search capabilities but has also resulted in real time search and indexing capabilities. This has made Cobot more robust, scalable and diverse in terms of the search features. In particular we have addressed issues such as mapping database entities efficiently to different semantically rich indices, quick mapping of entities from the database which is our primary persistent storage.

Zoie system has the following design properties:

1. New conversations are made available to searchers immediately

2. Indexing does not affect search performance

3. Additions of conversation does not fragment the index (which hurts search performance)

4. Deletes and/or updates of conversations does not affect search performance.

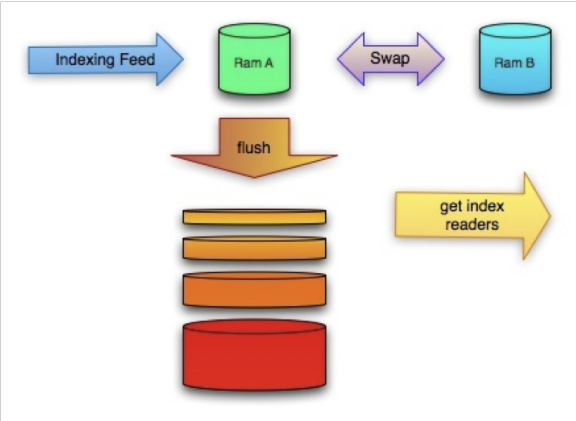

Figure 24: Zoie Architecture (source: zoie documentation)

Figure 24 above encapsulates the basic architectural crux of Zoie subsystem. A basic problem with normal indexing systems is that new document additions are not available for search/consumption immediately because there is a delay before the new document is added to the index. Zoie subsystem works around this problem, by pushing the new documents to an in memory index (RAM B as shown in figure) and simultaneously flushes the new document to the underlying disk based persistent index. The Zoie index reader reads data from both, the in memory index and the disk and presents the consolidated results to the search subsystem. For example, in the scenario above, the live indexing feed is first accumulated in memory in RAM A. The Zoie subsystem then swaps the content of RAM A and RAM B, thus making the newly added documents available to the index reader through RAM B. At the same time the data from RAM A is pushed to the disk index so that it is stored in a persistent manner. RAM B and the on disk index both act as datasources for the IndexReader.

CHAPTER IV

INFORMATION EXTRACTION

4.1 Introduction

Information Extraction (IE) is a generic term used for extracting structured content from text. Several text analytics tasks such as identifying noun phrases, facts, events, people, places and relationships are examples of Information Extraction tasks. These tasks are also called named entity recognition tasks that either use rule based approaches with thesaurus, regular expressions and grammars or probabilistic approaches. For IR and search applications, IE technologies are mostly used to identify contextually relevant features that involve text analysis, matching and categorization. Language technologies using part-of-speech tagging, etc. are applied to semantically annotate the documents with extracted features to aid search relevance.

The rapidly increasing volume of unstructured information poses the challenge of efficient and automated knowledge understanding so as to build computing systems that can acquire, represent, learn and maintain such knowledge, and efficiently reason from it to aid in knowledge discovery and re-use. The construction of these automated systems to assist decision making is impeded by difficulties in formalizing knowledge and in encoding that knowledge for use by computer agents that can integrate and reason from it.

4.2 Related Work

Information extraction(IE) has long been an active area of research in natural language field. One of the most challenging tasks of IE is to extract contextual meaning from sentences in documents and webpages and use it for problem solving tasks. IE

has applications in fields such as Question Answering where question focus is detected using different sophisticated algorithms and techniques. To extract relationships from sentences, one approach is to use templates that match specific linguistic structures. For example [115] utilizes templates to determine protein-protein interactions from biomedical literature. Machine Learning based approaches have also been utilized for extracting relations from unstructured text. For example, DIPRE [17] and Snowball [2] use bootstrapping, a general class of semi-supervised learning algorithms for extracting relations. On the other hand [120] utilizes fully-supervised learning methods for extracting relations. [77] gives a good overview of the Machine Learning based approaches for relation extraction.

Another challenge of information extraction systems is overcoming the performance bottleneck. These systems generally cannot afford to employ deep parsing technologies that generally require $\Theta(n^3)$ time algorithms where n is the length of sentence. Therefore it is not feasible to utilize deep parsing on large text corpus. To overcome this problem [4] presented a technique to query text databases to retrieve "promising" documents; the Information Extraction system processes only these documents.

Marti Hearst had suggested that hyponyms could be acquired from Large Text Corpora [56]. For example, consider the sentence *"The bow lute, such as the Bambara ndang, is plucked"*. Even if we have not encountered the terms *bow lute* and *Bambara ndang*, we can infer from the sentence that *Bambara ndang* is a kind of *bow lute*. Thus lexico-syntactic patterns can be utilized to discover information from a large Text corpus.

This technique has been successfully utilized to discover knowledge from the World-wide Web, the largest Text corpus available for machine processing today. Instead of gathering information from the Web directly, these systems utilize Web

search engines which have already crawled and indexed the information. For example, Know-it-all [45] was able to extract thousands of facts automatically using Web search engines. Similarly, PANKOW [33] could automatically discover names of resources like countries, cities and rivers. Several limitations of the PANKOW system have been alleviated by C-PANKOW [34].

Classification of terms is the determination of IS-A relation between the term and a class. Marti Hearst's idea has also been utilized to determine other type of relations including *part-of* [15] and causal [54]. In this paper we attempt to identify any arbitrary relation between two entities which is a much more challenging problem.

Techniques have also been developed for learning the patterns with which to query search engines. For example, [45] presents extensions to the Know-it-all system to improve its recall. In order to ensure that more terms can be correctly classified by querying WWW search engines, techniques like Rule Learning, Subclass Extraction and List Extraction were introduced. We have developed a technique for learning patterns for querying WWW search engines which is similar to the Rule Learning method; however we have generalized it for any type of relations. Our method is a bootstrapping based learning technique similar to DIPRE [17] and Snowball [2]. The main difference from the earlier systems is that we do not need to examine the full text to learn patterns for extracting relations; we just examine the snippets returned by the search engines.

Entity Annotation is a challenging research area that is precursor for efficient discovery of relations. Term extraction systems can be broadly divided into two types: those with a rule base and those with a learning method. In [52], protein names are identified in biological papers using hand-coded rules. On the other hand, in [35], supervised learning methods based on Hidden Markov Models are used. [107] have developed the BioAnnotator system, which is part of the current Relation Extraction system, and uses rules and dictionary lookup for identifying and classifying biological

terms.

Semantic Web [74] is a vision of the next generation World-wide Web in which data from multiple sources described with rich semantics is integrated to enable processing by humans as well as software agents. Semantic Webs are described using the Resource Description Format (RDF) language which provides a simple data model for describing relationships between resources in terms of named properties and their values. Resources can represent diseases, countries, companies, movies or any other entity or concept whose properties need to be represented semantically. RDF describes a Semantic Web using RDF *Statements* which are *triples* of the form <*Subject, Property, Object*>. Subjects are *resources*. Objects can be resources or literals. Properties are first class objects in the model that define binary relations between two resources or between a resource and a literal.

It is obvious that identifying relations between resources and describing them as RDF triples are essential initial steps to realize the vision of Semantic Web. However, the current situation of the Semantic Web is one of a vicious cycle wherein a true Semantic Web is non-existent due to the lack of a semantic markup of data, which in turn arises due to the difficulty of discovering and establishing relationships among concepts and resources existing on the Web.

One of the goals of Semantic Web research is to incorporate most of the knowledge of a domain in an ontology that can be shared by many applications. Various ontologies and knowledge bases have been developed for several domains For example *Unified Medical Language System (UMLS)* is a consolidated repository of medical terms and their relationships, spread across multiple languages and disciplines (chemistry, biology, etc). These ontologies organize information of various resources, each with their attributes, and describe simple relationships like *is-a* and *part-of* between concepts. However, they generally do not incorporate complex relationships between resources. For example, although UMLS contains details about many diseases, viruses

and bacteria, it does not incorporate relations between diseases and the causes of the diseases. Therefore, representing these ontologies and knowledge sources in Semantic Web ontology languages will not be sufficient to create a Semantic Web.

The World-wide Web today has become the most comprehensive repository of information. From the Web one can easily determine relations between resources for most domains. Thus one can determine the cause of typhoid, the capital of Fiji or the CEO of IBM. However, it is very difficult to utilize automated techniques to extract knowledge from unstructured Web pages. Moreover, because of the very large amounts of information, it is impossible to extract all these information manually and augment Semantic Web ontologies and knowledge bases.

There have been several related work on semantic role labeling [109], [86], [53], textual inference [40], [55], dependency parsing [41] and ontology alignment [62]. For example, the textual inference task is to determine if the meaning of one text can be inferred from the meaning of another and from background knowledge. Relationship Extraction system also apply heuristics, path learning and parsing techniques [120], [3]. The relationship extraction system aim at finding pre-determined paths and then apply a machine learning algorithm to learn such unseen paths. Besides knowledge intensive approaches relying on dictionaries and thesauri, techniques such as mutual information has been used previously [32] to identify collocations of terms for identifying semantic relationships in text. Our approach here uses simple yet high precision scoring functions for appropriate tree merging and creates a robust graph based infrastructure for semantic analysis and inference.

4.3 Entity Extraction

Entity extraction (also known as Named entity recognition (NER)) is a field of IE that seeks to locate and classify elements in text into different classes such as the names of

persons, places, organizations, locations, temporal quantities, drugs, diseases, treatments and such different semantic types. Entity extraction systems typically use linguistic grammar-based techniques as well as statistical models. Statistical model based classifiers typically require a large amount of manually annotated training data for high precision extraction.

4.3.1 Ontology based entity extraction

Unified Medical Language System (UMLS) is a system aimed to facilitate the development of computer systems that behave as if they "understand" the meaning of the language of biomedicine and health. It is developed by NLM for use by system developers in building or enhancing electronic information systems that create, process, retrieve, integrate, and/or aggregate biomedical and health data and information, as well as in informatics research. They can be used to support a range of functions involving one or more types of information, e.g., patient records, scientific literature, guidelines, and public health data. Since COBOT is a conversational system for enabling recommendations from the health domain besides others, UMLS is a quintessential component of the overall architecture. We have processed the UMLS Metathesaurus and the Semantic Net (Figure 25) with about a million terms and their semantic types into a fast searchable dictionary with support for approximate, skip based and overlapping matching strategies within noun chunks in sentences.

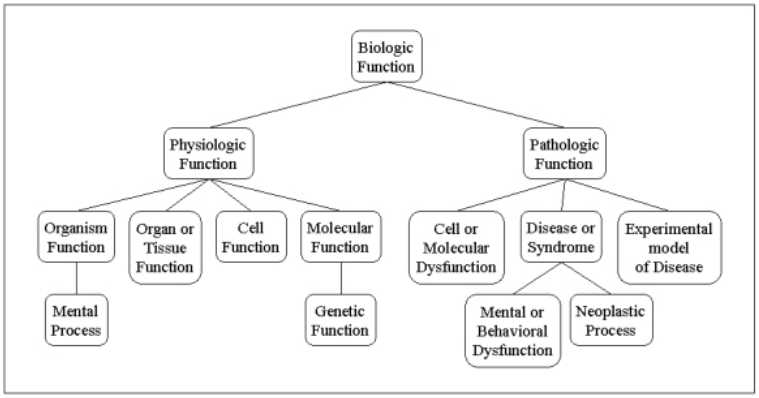

Figure 25: UMLS Semantic Network

ConceptMapper ([108]) is an open source tool for classifying mentions in an unstructured text document based on concept terminologies and yielding named entities as output. It is implemented as a UIMA (Unstructured Information Management Architecture ([48])) annotator, and concepts come from standardised or proprietary terminologies. ConceptMapper can be easily configured, for instance, to use different search strategies or syntactic concepts. In Cobot system, various NLP tasks such as Sentence splitting, Tokenization, POS tagging, Chunking, Relation extraction and Indexing are built as UIMA components. We have integrated ConceptMapper in Cobot's pipeline for the task of fast, accurate, multi-strategy entity extraction. MetaMap is a widely used medical entity extraction system from unstructured text in the medical domain released by the National Library of Medicine. ConceptMapper performance is comparable to MetaMap ([108]) but without the limitation of being tied solely to UMLS.

The way we have created medical concept mapping dictionaries is by loading more than a million UMLS Metathesaurus concepts with additional information such as Semantic Net types and inflectional variants. into XML dictionary files. ConceptMapper processes this large file and loads it in memory for matching and creating annotations. An entry in the dictionary files looks as follows:

```
<token canonical=‘‘Thrombocytopenia’’ SemClass=‘‘Disease or Syndrome’’
CUI=‘‘C0040034’’>
<variant base=‘‘Thrombocytopenias’’/>
<variant base=‘‘THROMBOCYTOPENIA NOS’’/>
<variant base=‘‘Thrombopenia’’/>
<variant base=‘‘Thrombopenias’’/>
<variant base=‘‘Thrombocytopenia, unspecified’’/>
</token>
```

Here, we see that the concept 'Thrombocytopenia' has a UMLS concept identifier

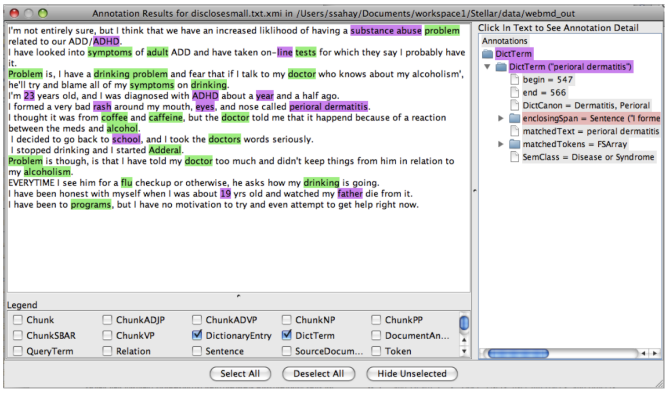

Figure 26: Concept Extraction example

(CUI) "C0040034", a semantic type "Disease or Syndrome" and different inflectional variants founds in various sources of literature.

Figure 26 shows some examples from the concept extractor for the document.

4.3.2 Tag based entity extraction

Many sites such as online bookmarking sites, blogging sites and Q&A sites with user-generated content allow users to tag their content. Tags are freely chosen keywords that assign topics and subjects to content. Sites also use tags to organize and provide access to content. Stack Overflow is one such social Q&A site that makes all information gathered in the system publicly available every month. Every question must be assigned at least one tag on the site. The dataset we used contains about 25,000 users assigned tags on programming, computer science and mathematics topics. We used this set for matching and annotating entities from computer science and mathematics domain for our conversational system. This set does not give us the richness of information we get in medical entity extraction using a vast curated resource such as UMLS. One of our purpose was to compare how our medical conversational information access system performed compared to educational conversational information access system.

To check the coverage of the Stack Overflow tags on conversations related to CS

topics, we did a small experiment. We randomly selected 244 posts from a web forum containing CS related conversations. We were able to tag 209/244 of these conversations with at least 1 tag from our tag set. Some examples of the tags we were able to assign to posts were as follows (tags shown at the end of posts in brackets):

```
hello world works , lists functions work but for loops  freeze help.
[ lists,  functions, loops]
 Please unsubscribe openstudy@gtod.net from receiving the digests.
 I have tried 3 times to do this using the instructions at the bottom of
 the digest however it does not work.  Many thanks, Greg.
  [ digests, times,  using,  instructions,  bottom,  digest]
Hi, what courses should I choose to work as Database professional.
Sorry for my English.
[ courses, Database,  English]
How do you know which readings go with which lectures? All the
reading says is lecture 1-3  and has a list of 4 readings but are you
supposed to read one before or after every lecture?
Are supposed to read all before or all after what do you do?
[ readings,  lectures,  reading,  list, lecture]
Can anyone pls tell me where can i down load the platform that will
allow me to use python pls .
[ load,  platform, python]
```

After studying the conversations we weren't able to tag, we saw that many of the conversations did not contain any tags. Some example of these untagged conversations are as follows:

```
Anyone looking at problem set 3?
Problem set/exam questions
```

```
http://man.lupaworld.com/content/develop/Advanced_Python_programming.pdf
Is there a way to get the handouts?
Please excuse because this question has been asked before. But,
is there some way to get the handouts for this 6.0 course?
A lot of guests here. Where are you studying from?
How can we improve OpenStudy? Tell us here when you have a moment!
Help with PS2 nuggets code?
How can I consolidate a Tuple e.g. make (1,5,4,6,5,2,5,1) just (1,2,4,5,6)
How to code?
I'm really close with ps1a but can't work out where I'm going wrong.
where do I start?
what is 1+1?
```

4.3.3 Keyword based entity extraction

We also used a fallback strategy for identifying potentially important keywords in conversations when we weren't able to extract entities using UMLS or Stack Overflow tags. While processing the sentences in our annotators, we were 'normalizing' the sentences by removing duplicates, stopwords and function word tokens and picking up nouns from these strings and storing these keywords in our database. Function words are words that have little lexical meaning (for example, articles, pronouns, conjunctions, etc.), they instead serve to express grammatical relationships with other words within a sentence, or specify the attitude or mood of the speaker. These words are generally filler words that help in forming grammatically correct sentences.

This step helps us in creating a gracefully degrading conversational information access system. We do not use the extracted keywords from this step in other upstream IE tasks such as relationship and query candidate extraction but use it to create some keyword based queries for search engines and our semantic search index.

4.4 Relationship Extraction

In this section, we introduce techniques of identifying relations between resources. We utilize a Web search engine to first determine Web pages that have those relations. Our system is very efficient because instead of downloading these documents we only process the result snippets. Since these snippets are only one or two sentences, information extraction is much simpler.

We first query the search engines with lexico-syntactic patterns to retrieve relevant information. These patterns are initially hand-crafted but can be progressively learnt. Instead of downloading Web pages, we extract relations from snippets, the small section of the result pages that contain relevant text from the search results containing the query string. The knowledge discovered by this technique can also be used to augment the ontologies and knowledge bases and create a Semantic Web of a specific conceptual domain. As an experiment, we have utilized the technique to discover relations from general biomedical field to a specialized biomedical sub-domain for domain specific ontology construction. Our experiments show the promise of our technique.

In this section we will explain our technique utilizing the search engine results to discover relationships between resources. We also describe how the discovered relations can be used to augment Semantic Web ontologies and knowledge bases and a method of learning the patterns to query search engines in [94].

4.4.1 Relation Identification

Let us assume that our objective is to discover causal relationship between a disease and a biomedical entity. Given a disease d and a biomedical entity e, we can query search engines with phrases like *"e causes d"* or *"d is caused by e"* and count the number of results that are retrieved. However, there are thousands of entities (viruses, bacteria, parasites, etc.) that can cause a disease. Querying for each of them is not

```
relationIdentifier(resource,property) {
   patterns = List of patterns in the Pattern Database for property
   synonyms = List of synonyms in the Ontology for resource
   initialize a Hash Map resultResources

   for each s in synonyms {
      for each p in patterns {
         queryString = p with ‘‘RESOURCE’’ replaced by s
         results = SearchResultSnippets(‘‘queryString’’)
         for each result in results {
            parsedResult = PoSTag(result)
            entityAnnotatedResult = EntityAnnotate(parsedResult)
            relAnnotatedResult = RelationAnnotate(entityAnnotatedResult)
            resultResource = relationEntity(relAnnotatedResult,s)
            resultResources{resultResource}++
         }
      }
  }
  return resultResources
}
```

Figure 27: Pseudo code to determine the entity that has the relations specified by *property* with *resource*

efficient. It would be more useful if given a disease we can discover the likely causes of the disease.

We have implemented a generic framework for discovering relations between resources. Figure 27 shows the pseudocode to determine the entity that takes part in relations specified by *property* with *resource*. For each property, patterns that indicate each of these relations are manually determined and entered in a *Pattern Database*. Example pattern for the property 'cause' is as follows:

- **causes**: causes RESOURCE, RESOURCE is caused by

More common patterns that occur on web databases between resources can be learnt by our Pattern Learner module to augment the Pattern Database. This is discussed in Section 3.3.

We also determine synonyms for the given resource using an ontology. For each synonym and each pattern we issue phrase queries to a Search Engine. Presently we utilize Google WWW search engine and Pubmed database search engine. Thus if we want to determine p53 gene effectors, we would issue queries like *"p53 is affected by"*, *"affects p53"*, *"bears on p53"*, *"impacts p53"*, etc. We are using WordNet and UMLS ontologies for this purpose.

Previous systems like Know-it-all [45] and PANKOW [33] classify entities by counting the number of results retrieved by Google. Unfortunately, just the number of results is not sufficient for our purpose. However, downloading the result pages will make the process very slow. Therefore, we utilize the *result snippets*, the small section of the result pages that contain the query string that is returned with a Google search and *abstract titles*, that are returned by the Pubmed web services search calls.

We determine the resource that is related to the given *resource* from these result snippets using 3 components:

- We first parse the snippet using a *Part-of-Speech Tagger*. This identifies entities

like Noun Phrases, Verb Groups, etc.

- Then an *Entity Annotator* determines the resources (or entities) in the strings using ontologies as well as a Rule Engine. If all the synonyms of a resource are specified in an ontology, the Entity Annotator can identify a resource in a snippet in spite of variations in its naming. In many cases the ontology may not be comprehensive and may not contain all possible resources. In that case our Entity Annotator can recognize names of entities like variations of ontological terms not present in the ontology, Chemicals, etc. using a Rule Engine.

- Finally a *Relation Annotator* discovers the relations between the resources. At present we are using a simple template-based technique for relation identification. For example, some common templates which specify relationships in sentences are:

  - *Subject Verb_Group Object* (For example, *"HIV causes AIDS"*)
  - *Object Passive_Verb_Group Subject* (For example, *"AIDS is caused by HIV"*)
  - *Noun (Nominal form of verb) Object Subject* (For example, *"causing of AIDS by HIV"*)

- If a template is matched it is assumed that a relation of the matching verb group (or nominal form) has been identified. Note that if there are noun phrases or adjectives between the entities and the verb groups in the sentences they are considered as qualifiers for the result resource. We have avoided using a deep parser as it considerably slows down the relation identification process for relation triples. Identification of complicated relations in longer sentences would deeply benefit from using a dependency parser.

The combination of Entity Annotator and Relation Annotator creates an annotated string from which the entity taking part in the relation with the *resource* can

be easily identified. For example given the result snippet *"AIDS is caused by HIV"*, the Part-of-speech tagger will recognize *"is caused by"* as the Verb Group, Entity Annotator will recognize *AIDS* and *HIV*, and the Relation Annotator recognizes *HIV* as the resource that is in causal relationship with *AIDS*. On the other hand for the more complex result snippet *"Metabolic bone disease is caused by the lack of Vitamin D3"*, the Relation Annotator recognizes *"Vitamin D3"* as the resource that is in causal relationship with *"Metabolic bone disease"* with the qualifier *"the lack of"*.

Different authors will express the same semantics in different ways. Therefore there will be variations in the results that are retrieved by search engines. For example, one result may state that *AIDS* is caused by *HIV* while another may state that the disease is caused by *Human Immunodeficiency Virus*. However, Entity Annotator will map them to the same resource using ontologies. Therefore, Relation Annotator will identify the same biological entity from the two search results. However, this may not be true for all snippets. For example, if one snippet states that *Metabolic bone disease* is caused by *"the lack of Vitamin D3"* and another states that it is caused by *"Calcium deficiency"*, our annotators will not be able to match the two entities. Therefore, as shown in Figure 27, a hash map that has the resources that have the specified relation with the given resource along with the number of occurrences for each of them are returned from the *relationIdentifier* procedure.

The Relation Identifier returns a hash map containing the potential relation resources along with the count of the number of snippets that contain the relation. We have utilized our technique to identify various types of relationships between resources. However, a formal evaluation of our technique is difficult because there are no test data sets that can be used for the evaluation. For determining the efficiency of our technique, we determined relations for resource for various domains relevant to the UMLS knowledge base. In the absence of domain experts, we did literature surveys and Web surfing to determine whether the relations identified by our system

are correct. We report some of our past results here[79].

For UMLS besides Semantic Network properties *causes*, *diagnoses*, *consists of* and *affects* we also extracted *binds* relations for several entities of UMLS class *Amino Acids, Peptides or Proteins*. Table 1 shows several biomedical relations determined by our technique. Thus we could identify the cause of *Thyphoid* (*Bacterium Salmonella Typhi*) as well as entities that affect *Statin* (*Lipitor, Gemfibrozil, Niaspan*).

Table 1: Some relations for UMLS resources determined by our technique

Property	UMLS Resource	Relation Resource
causes	Typhoid	Bacterium Salmonella Typhi
diagnoses	Cyst	Ultrasonography
consists of	Butane	Liquefied Petroleum Gas
affects	Statin	Lipitor, Gemfibrozil, Niaspan
binds	Rhodopsin	Lys296, Transducin

Table 2: Coverage and Correctness of the Relation Identifier for UMLS Resources

Property	Class	Coverage	Correctness
causes	Disease	0.85	0.82
diagnoses	Anatomical Abnormality	0.9	1.0
consists of	Organic Chemical	0.72	0.75
affects	Gene	0.76	0.8
binds	Amino Acid, Peptide, or Protein	0.75	0.83

Table 3: Some retrieved relations for UMLS resources from Pubmed

Property	UMLS Resource	Relation Resource
causes	Typhoid	Salmonella enterica serotype paratyphi
diagnoses	Cyst	Hypophysitis, Ciliary body melanoma
consists of	Butane	null
affects	Statin	Cholesterol, Angiogenic mediators
binds	Rhodopsin	Arrestin

Table 4: Coverage and Correctness of the Relation Identifier for UMLS Resources using Pubmed search

Property	Class	Coverage	Correctness
causes	Disease	0.9	0.88
diagnoses	Anatomical Abnormality	0.8	1.0
consists of	Organic Chemical	0.5	1.0
affects	Gene	0.8	1.0
binds	Amino Acid, Peptide, or Protein	0.8	1.0

For each property, we determined relations for several entities of some particular UMLS class which has that property. To test the system impartially we have included common as well as rare concepts in our experiments. We calculated the following statistics for each property from our experiments:

- **N**: Total number of resources for which we tried to identify relations.
- **F**: The number of resource for which at least one relation was identified by our system.
- **C**: The number of resources for which at least one relation that was identified by our system is correct.
- **Coverage (CV)** $CV = \frac{F}{N}$

- **Correctness (CR)** $CR = \frac{C}{F}$

While *Coverage* measures the number of relations for which results could be obtained from the search engines, *Correctness* measures the ability of extracting the correct relation resource from the returned results. These metrics resemble recall and precision used in Information Retrieval. It is difficult to calculate exact recall as Google limits the searches to 10 results per search query and we are only processing the top 100 results returned from Pubmed searches. Table 2 shows the results for each property and the corresponding UMLS class retrieved from Google search engine. Table 4 shows the results for each property and the corresponding UMLS class retrieved from Pubmed database search.

There are several observations when we compare the results of Pubmed and Google searches. Medline abstracts are precise and technical accounts of facts and experiments reported through literature. They assume prior contextual knowledge and are highly domain specific in nature. We observed that they fail to identify the common answers to our queries as returned by Google search. However, they pick up some answers that are rare and can only be found through scientific papers. A good scheme of relation extraction aimed towards ontology construction would be to combine both these techniques to find common as well as rare relations for domain specific searches.

4.4.2 Quality of the Relation Identifier

The quality of the Relation Identifier is affected by various factors:

- The coverage is affected by Google's inability to identify complex class associations such as chemicals, genes, proteins and their relationships. For example, Google is unable to retrieve any results on our queries such as *"binds Auxin Response Factor 1"* or *"Nephroptosis is diagnosed by"*.

- Sometimes the snippet returned by Google may not be able to identify the resource property. For example, one snippet retrieved was *"Primary Hypertension*

is caused by abnormalities of" with the relevant cause of the disease stripped off.

- The Relation Identifier fails to identify complex relations embedded in large sentences or spanning multiple sentences (coreference and anaphora resolution).
- Setting a high threshold on relation identifier retrieves high precision results at the cost of recall.

Our experiments show that UMLS is really comprehensive and has all biomedical resources and its variations. Therefore our system should be integrated with systems that identifies and classifies entities in Web pages like Know-it-all, PANKOW and SemTag [42] to create a comprehensive Semantic Web.

Some examples of relations with categorized links:

- Dipyridamole "AFFECTS" platelet thrombus growth
- Adenosine "BRINGS ABOUT" catecholamine
- Myocardial ischemia "INDICATES" stellectomy
- Ischemic complications "COMPLICATES" coronary angioplasty
- Ischemia "DISRUPTS" neuronal cytoskeleton
- Tc-99m sestamibi "EXHIBITS" parathyroid disease
- Dipyridamole "MEASURES" methotrexate

4.5 Augmented Transition Network

In this section, we will describe our top-down backtracking search based parsing algorithm based on Augmented Transition Network to extract candidate query phrases from sentences. An augmented transition network is a directed graph in which parsing is described as the transition from a start state to a final state in a transition

```
Step 1: Set the ATN pointer to the start state and the sentence pointer
     to the beginning of the sentence being parsed.
Step 2: Search through the arcs leaving the node until one is found
     whose test succeeds. In order to legally traverse the arc, it must
     satisfy the following conditions:
     Any associated test must be satisfied by the current variable
     values.
     If the arch is labeled with a word category (e.g., noun) the
     current word must be a member of that category.
Step 3: Execute the actions associated with the arc. In addition, do the
    following, based on the type of arc:
    If the arc is a word category, increment the current position and
    change the current node to the destination of the arc.
    If the arc corresponds to another ATN, push the starting node of
    that ATN onto the ATN pointer.
    If there are no more arcs, pop the current node off of the ATN
    pointer and set * to the value returnedby this node. Succeed if
    ATN pointer is now empty and all of the text has been processed.
    If the ATN pointer is empty and text remains, fail. Otherwise, return to
    step 2.
```

Figure 28: ATN Parsing algorithm

network corresponding to an English grammar [116]. The nodes represent states in the parse; each arc contains a test which must succeed for the arc to be traversed. If the arc is traversed, an action is performed. The parse proceeds by means of a depth-first search of the ATN; it succeeds if no more arcs are to be followed and end of input is reached. ATN was first used in LUNAR system, one of the first question answering systems [117]. An ATN is similar to a Finite State Machine in which labels or arcs between states can be calls to other machines. Arcs in an ATN may contain words, categories (e.g., noun phrase), they may push to other networks, or perform procedures.

The basic bottom-up ATN parsing algorithm 28 is described as follows:

In the following figures 29 30 and 31, we show our backtracking search ATN machines that try to extract longest possible Concept-Verb Phrase - NounPhrase

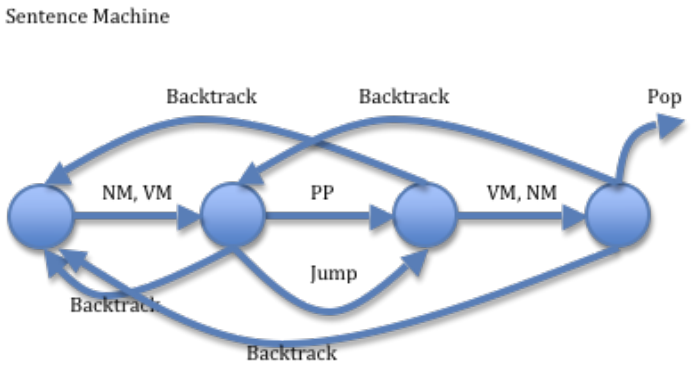

Figure 29: ATN Machine

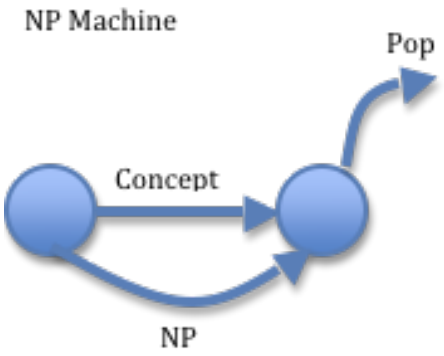

Figure 30: Noun phrase machine

and other similar rules from the sentences. If the sentence machine doesn't find any matching rule with all backtracking, it proceeds the word counter and starts from the initial stage.

Figure 32 shows some extracted phrases using the ATN machines. These phrases are: 'helping your ADHD', 'treating adult ADHD and substance abuse problems', etc.

4.6 Query Transformation Strategies

Cobot uses a mix of strategies in it's knowledge goal and task goal for generating queries from conversations. The generated queries are sent to retrieval engines to generate a candidate pool of recommendations for upstream filtering processes. There are two query generation systems in Cobot, one for retrieving results from the web

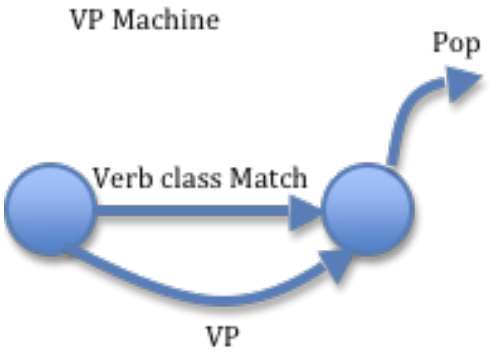

Figure 31: Verb Semantic class matching machine

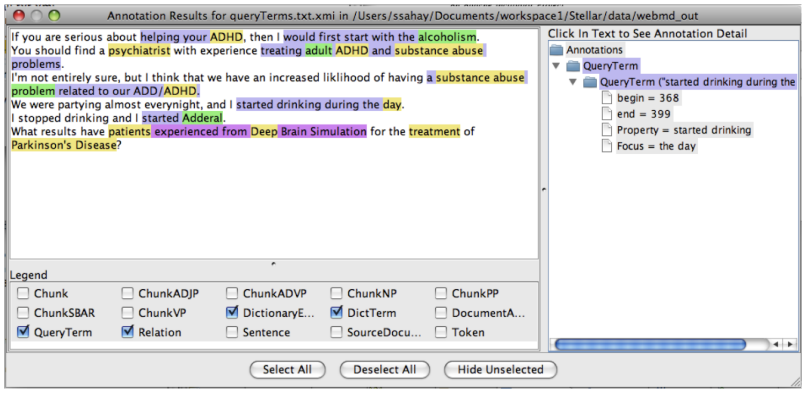

Figure 32: Query Candidate examples

and another for retrieving results from Cobot's semantic search index that contains current (results retrieved for the current conversation) as well as past results from recommendations generated from community conversations. Cobot's knowledge based goal is to recommend specific content giving pointers to answers and related support and validation sources of information. Cobot's task goal is to recommend learning resources providing definitions, facts, methods, tutorials and other learning resources from informational pages, forums and Q&A sites.

The kinds of queries in Cobot are as follows:

- *Keyword based queries*: These are Boolean OR queries on extracted keywords in text after text processes such as stop word removal, duplicate word removal, and selection of nouns and verbs.

- *Concept based queries*: These are Boolean OR queries on domain specific extracted concepts in text. Cobot, in it's knowledgebase contains concepts extracted from UMLS medical ontology and a generic tags based vocabulary extracted from StackOverflow data consisting Computer Science and Math related conversations.

- *Semantic Class based queries*: When Cobot has knowledge about the concept's semantic classes (categories), it creates Boolean OR queries consisting of Semantic Classes in conversations.

- *Concept-Concept Relation queries*: Cobot extracts relationships in between concepts in conversation and uses it as a query for searching

- *Complex Relation queries*: Cobot extracts complex relationships between nouns and concepts in conversations using an Augmented Transition Network with rules to identify sentence and phrase focii and their property. Cobot also uses these relationships as queries.

```
Step 1: analyze:
          get keywords, concepts, semantic types, relationships,
              key phrases with focus and their properties and speech
              acts of sentences in conversation C
Step 2: filter, augment, normalize
    get Question, Advisement and Disclosure sentences S in C
    for each sentence s in S:
      extract Complex Relation queries if they exist
     else extract concepts and augment them based on task
goals using WordNet expansions and custom rules
else use normalized keywords as query for short sentences
Step 3: generate Web Search Query and forward them to custom
          search engines (the retrieved results are semantically indexes
      in the local search indices)
Step 4: generate Lucene queries (phrase based, span based,
             keyword based) on semantic fields for the local community
              specific search index
Step 5: search the index using queries on the keywords, concepts,
              semantic types and relationship fields in the indexed documents.
Step 6: send results to candidate filtering engine for post-retrieval
          recommendation generation
```

Figure 33: Query Processing

For searching the web for candidate recommendations, Cobot combines all of the above queries as one large Boolean OR query and retrieves results from web search engine thus avoiding multiple wire requests.

Figure 33 show the high level query generation and retrieval process in Cobot.

We broadly categorize query processing in Cobot at a strategic level as follows: Ṡtrategies have been.

1. Recognize and retrieve - Cobot's concept mapping engine maps lexical variants of spans of strings in sentences and chunks of phrases to one of the millions of concepts loaded in Cobot's memory at startup time.

2. Query Augmentation - Cobot tries to decipher the kinds of tasks user is interested in when asking or replying in a conversation. Cobot uses this task knowledge to augment queries with words and categories. Such functionality

tries to ensure relevant types of documents are retrieved and presented appropriately. One successful application of this type of strategy has been discussed in[20].

3. Translation - Several normalization rules have been applied in Cobot with the goal to represent text in standardized form. Rules such as stem, strip stop words, convert short form phrases to regular form (for example, 'I've' to 'I have'), remove function words using a custom function word dictionary.

4. Source specific transformations - Conversations from different domains are routed to custom sources, repositories and indexes (for example, a medical query is sent to a custom medical search engine and queries from a math conversation is sent to the math based semantic index)

CHAPTER V

INFORMATION RETRIEVAL

Information Retrieval(IR) methods have made significant advances in the last sixty years of significant research progress and commercial breakthroughs. The mature, yet simple, reliable IR technology built on the notion of taking words as they stand along with the frequencies has a few important modeling lessons for natural language AI technologies. In IR, words are the atomic units of representations whereas AI leans towards representing words with more formal logical knowledge representation formalism. In any case, human information needs are vague and relies on the notion of relevance to context rather than exactness. To address this vagueness of human information need, AI information processing techniques need to build in synchrony with IR technology and add semantic layers and different knowledge goals for intelligent information processing. IR technology directly addresses the acquisition bottleneck problem; once we address this issue, we can apply information extraction, reasoning and learning algorithms to build solutions for real world complex problems.

Natural language understanding systems perform subjective analysis of the input text based on tasks determined by different knowledge goals([89]). For example, a text analytics task involves syntactic and semantic processing; a memory level task involves recognition, classification and generalization; a explanation based task involves determining cause and effect and a relevance based task involves identification of aspects of current situation. Cobot system is primarily concerned with handling tasks for the relevance goals or user and system.

Three components are closely inter-twined in an information retrieval system. These are:

1. the text describing the information need

2. the set of documents relevant to the topic

3. entire collection of documents

In this chapter, we will briefly review Information Retrieval methods and processes, related work and how we are using IR in Cobot involving the above three components to get good retrieval performance.

An IR system can be characterized at different levels by types of users, types of data and the types of information need, along with the size and scale ofthe information repository it addresses. Different IR systems are designed to address specific problems that require a combination ofdifferent characteristics.These characteristics can be briefly described as follows:

Types of Users: The user may be an expert user (e.g., a curator, a librarian), who is searching for specific information that is clear in his/her mind and forms relevant queries for the task, or any layman user with a generic information need. The latter cannot create highly relevant queries for search (e.g. students trying to find information about a new topic, researchers trying to assimilate different points of view about a historical issue, a scientist verifying a claim by another scientist, a housewife trying to shop for clothing).

Types of Data: Search systems can be tailored to specific types of data. For example, the problem of retrieving topical information may be handled more efficiently by customized search systems that are built to collect and retrieve only information related to a specific topic. The information repository could be hierarchically organized based on a concept or topic hierarchy. Domain specific or vertical IR systems are not as large or as diverse as the generic World Wide Web. Given that these specific collections exist or that they are acquired through a knowledge acquisition process, they can be exploited much more efficiently to respond to different kinds of

queries posed by the user for effective retrieval.

Types of Information Need: In the context of Web search, users information need may be defined as navigational, informational or transactional. The purpose of navigational search is to reach a particular piece of information (such as Georgia Techs website) that a user needs quickly. The purpose of informational search is to find static information (such as research activities at the college of computing, Georgia Tech) about a topic (the classic IR system task). The purpose of transactional search is to reach a site where further interaction happens (such as joining a social network, product shopping, online reservations, accessing databases, etc.)

Levels of Scale: In the words of Nobel Laureate Herbert Simon , "What information consumes is rather obvious: it consumes the attention of its recipients. Hence a wealth of information creates a poverty of attention, and a need to allocate that attention efficiently among the overabundance of information sources that might consume it." This overabundance of information sources in effect creates a high noise-to-signal ratio in IR systems. Especially on the Web, where billions of pages are indexed over distributed systems, IR interfaces are built with efficient scalable algorithms for distributed search, indexing, caching, merging and fault tolerance. Enterprise search systems offer IR solutions for search of different entities within the intranet of an enterprise such as emails, corporate documents, manuals, charts, presentations, and reports related to people, meetings and projects. They still typically deal with hundreds of millions of entities in large global enterprises. At even a smaller scale, there are personal information systems such as those on desktops, called desktop search engines, for retrieving files, folders and different kinds of entities stored on the computer (e.g., Google desktop). There are peer-to-peer systems such as BitTorrent allowing sharing of music in the form of audio files and there are specialized search engines for audio such as Yahoo audio search and Lycos audio search.

5.1 History of Information Retrieval

Man has been performing information retrieval as a common task for several centuries. It dates back to the times of ancient civilizations that devised ways to organize, store and catalog books and records. It has been the result of efforts that allowed knowledge to be retained and transferred from generations to generations. With the emergence of public libraries and printing press, large scale production, collection, archival and distribution methods evolved. With the advent of computers and automatic storage systems, the need for these methods to be replicated and applied to computational systems was realized. Several works emerged in 1950s such as the seminal work of H. P. Luhn where he proposed using words and their frequency counts as indexing units for documents and measuring word overlap as the retrieval criterion. It was soon realized that storing large amounts of text was easier the harder task was to retrieve that information selectively to users who wanted to access them. Methods that explored word distribution statistics gave rise to the choice of keywords based on their distribution properties , and keywords based weighting schemes. The earlier experiments with document retrieval systems such as SMART in 1960s adopted inverted file organization based on keywords and their weights as the method of indexing. Serial organization proved inadequate if queries required fast near real time response times. Proper organization of these files became an important area of study; document classification and clustering schemes ensued. The scale of retrieval experiments remained a challenge due to lack of availability of large text collections. This soon changed with the World Wide Web. Also, the Text Retrieval Conference, or TREC was launched by NIST (National Institute of Standards and Technology) in 1992 as a part of the TIPSTER program with the goal of providing a platform for evaluation of information retrieval methodologies and to facilitate technology transfer to develop IR products.

A search engine is a practical application of information retrieval to large scale

document collections. With significant advances in computers and communications technologies, people today have an interactive access to enormous amounts of user generated distributed content on the WWW. This phenomenon has instilled growth in search engine technology. where these engines are trying to crawl different kinds of real time content found on the Web. Other types of search engines include the desktop and the enterprise. For example, the biomedical literature search database was started in the 70s and is now supported by the Pubmed search engine which gives access to over 20 milllion abstracts.

While continuous progress is being made to tailor search results to the needs of an end user, the challenge remains in providing high quality, pertinent and timely information that is precisely aligned to the needs of individual users.

5.2 Information Retrieval Pipeline

The focus of IR is on retrieving documents based on the content of their unstructured components. Most documents are made up of unstructured natural language text composed of character strings from English and other languages. Common examples of documents include newswire services (e.g., AP or Reuters), corporate manuals and reports, government notices, Webpage articles, books and journal papers. There are two main approaches to IR statistical and semantic. In a statistical approach, documents are broken down into chunks of texts (words, phrases or n-grams) where each word or phrase is counted, weighted and measured for relevance or importance. These words and their properties are then compared with the query terms for potential degree of match to produce a ranked list of resulting documents that contain the words. Statistical approaches are mainly classified into one of the following approaches: Boolean, vector space and probabilistic. Semantic approaches to IR use knowledge-based techniques of retrieval that broadly rely on the syntactic, lexical, sentential, discourse-based and pragmatic levels of knowledge understanding. In

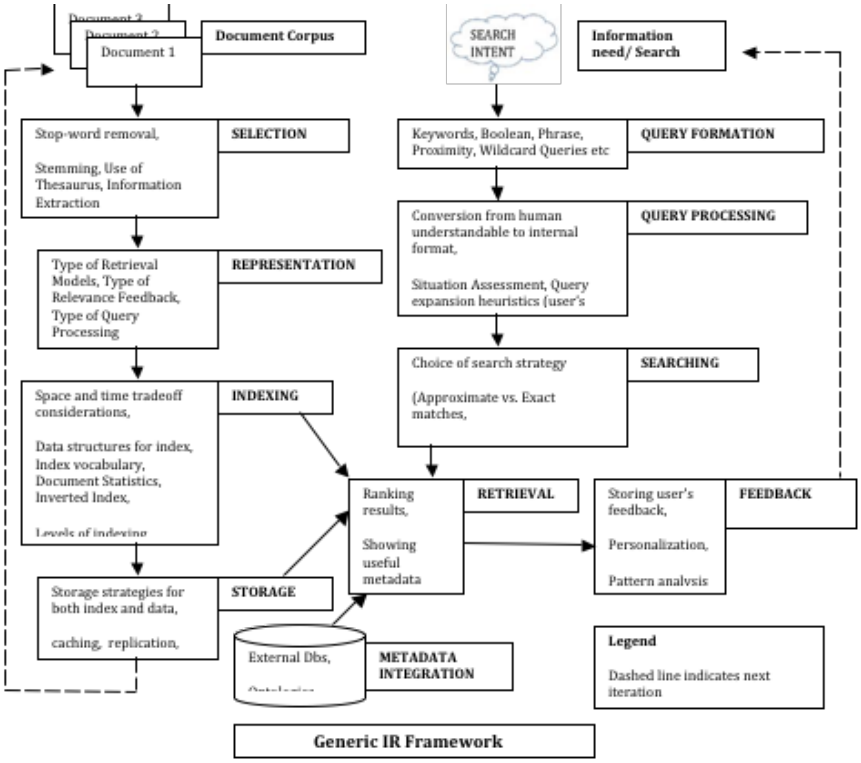

Figure 34: Generic IR Framework

practice, semantic approaches also apply some form of statistical analysis to improve the retrieval process.

In Figure 34, we show the various stages involved in the IR processing system. The steps involved for document pre-processing, document modeling, and indexing are shown on the left. These are typically off-line processes which prepare a set of documents for efficient retrieval. The steps involved in query formation, query processing, searching mechanism, document retrieval and feedback collection are shown on the right. In each of the box we have attempted to highlight the important concepts and issues. The rest of this introductory chapter is devoted to describing most of the concepts involved in the various tasks mentioned within the IR process shown in this figure 34.

5.3 Retrieval Models

As we noted in the previous section, the challenges of IR systems are centered around development of techniques for efficient and precise retrieval of relevant information aligned to a users information need. The retrieved set of results of an IR task is presented to the user as a ranked list of documents. The ranking algorithms work according to different notions of representation and relevance. These notions and assumptions about how to represent a document and how to judge its relevance to a user query are captured and formalized in different models (refer to Modeling task in Figure XX.1) that form the basis of search in IR. We briefly describe the important models of IR in this section.

5.3.1 Boolean Model

In this model, documents are represented as a set of terms. Queries are formulated as combination of terms using the standard Boolean logic set-theoretic operators such as AND, OR and NOT. Retrieval and relevance are considered as binary concepts in this model, i.e., the retrieved elements are an exact match retrieval of relevant documents. There is no notion of ranking of resulting documents. All retrieved documents are considered equally important - a major simplification in this model without taking into consideration frequencies of document terms or their proximity to other terms compared against the queries.

Boolean retrieval models lack sophisticated ranking algorithms and are amongst the earliest and simplest information retrieval models. It is also easy to associate metadata information and write queries that match the contents of the documents as well as other properties of documents such as date of creation, author, type of document, etc.

5.3.2 Vector Space Model

The vector space model proposes a framework in which term weighting, ranking of records and relevance feedback are possible. Documents are represented as features and weights of term features in an n-dimensional vector space of terms. The process of selecting terms and their properties as a sparse list of very high dimensional vectors (the vocabulary can contain hundreds of thousands of terms) is independent of the model specification. The query terms vector is compared to the document vectors for similarity/relevance assessment. The similarity assessment between two vectors is not inherent to the model; however, cosine of the angle between the query and document vector is commonly used for similarity assessment. As the angle between the vectors decreases, the cosine of the angle approaches one, meaning that the similarity of query with a document vector increases. Terms are weighted proportional to their frequency counts to reflect importance of terms in the calculation of relevance measure. To contrast, Boolean model does not take into account the frequency of words in the document for relevance match. This Document term weight wij (for term i in document j) is represented based on some variation of TF (term frequency) or TF-IDF (term frequency - inverse document frequency) scheme (as described below). TF-IDF is a statistical weight measure that is used to evaluate the importance of a document word in a collection of documents.

$$cosine(d_j, q) = \frac{< d_j \times q >}{\|d_j\| \times \|q\|} = \frac{\sum_{i=1}^{|V|} w_{ij} \times w_{iq}}{\sqrt{\sum_{i=1}^{|V|} w_{ij}^2} \times \sqrt{\sum_{i=1}^{|V|} w_{iq}^2}} \quad (1)$$

In the formula given above, we use the following symbols:

d_j is the document vector.

q is the query vector

w_{ij} is the weight of term i in document j

w_{iq} is the weight of term i in query vector q

$|V|$ is the number of dimensions in the vector that is the total number of important

keywords (or features).

TF-IDF uses the product of normalized frequency of a term i (TF_{ij}) in document D_j and the inverse document frequency of the term i (IDF_i) to weight a term in a document. The idea is that terms that capture the essence of a document occur frequently in the document (that is, their TF is high), but if such a term were to be a good term that discriminates the document from others, it must occur in only a few documents in the general population (that is, its IDF should be high, as well).

IDF values can be easily computed for a collection of some fixed size documents. In case of Web search engines, taking a representative sample of documents approximates IDF computation.

$$TF_{ij} = \frac{f_{ij}}{\sum_{i=1}^{|V|} f_{ij}} IDF_i = \log \frac{N}{n_i} \tag{2}$$

In these formulas, the meaning of the symbols is:

TF_{ij} is the normalized term frequency of term i in document D_j.

f_{ij} is the number of occurrences of term i in document D_j

IDF_i is the inverse document frequency weight for term i.

N is the number of documents in the collection

n_i is the number of documents in which term i occurs.

Note that if a term i occurs in all documents, then n_i =N and hence $IDF_i = \log(1)$ becomes zero nullifying its importance. Since the weight of term i in document j, wij is computed based on its TF-IDF value in some techniques, to prevent division by zero they typically add a 1 to the denominator in formulae such as the cosine formula above .

The Rocchio[95] algorithm is a well-known relevance feedback algorithm based on the vector space model that modifies the initial query vector and its weights in

response to user identified relevant documents. It expanded the original query vector q to a new vector q_e as follows:

$$q_e = \alpha q + \frac{\beta}{D_r} \sum_{d_r \in D_r} d_r - \frac{\gamma}{D_{ir}} \sum_{d_{ir} \in D_{ir}} d_{ir} \tag{3}$$

Here, D_r and D_{ir} are relevant and irrelevant document sets and $\alpha, \beta and \gamma$ are parameters of the equation. The values of these parameters determine how the feedback affects the original query and these may be determined after a number of trial-and-error experiments.

5.3.3 Probabilistic Model

Boolean and vector space models make implicit assumptions about representations and relevance that lead to the development of ad hoc similarity measures. For example, the vector based retrieval model assumes that documents closer to the query in cosine space are more relevant to the query vector.

In the probabilistic model, a more concrete and definitive approach is taken. The obvious order in which to present documents to the user is to rank documents by their estimated probability of relevance with respect to the query and the document. This is the basis of the Probability Ranking Principle due to Robertson :
To retrieve relevant documents in the probabilistic framework, the IR system has to decide whether the documents belong to the relevant set or the non-relevant set for a query. In order to make this decision, it is assumed that we have a relevant set and a non-relevant set and the task is to calculate the probability that the document belongs to the relevant set and compare that with the probability that the document belongs to the non-relevant set.

Given the document representation D of a document, estimating the relevance R and non-relevance NR of that document involves computation of conditional probability P(R—D) and P(NR—D). These conditional probabilities can be calculated

using Bayes Rule:

$$P(R|D) = P(D|R) \times P(R)/P(D) P(NR|D) = P(D|NR) \times P(NR)/P(D) \quad (4)$$

We classify the document D as relevant if P(R—D) ¿ P(NR—D). Discarding the constant P(D), this is equivalent to saying that a document is relevant if:

$$P(D|R) \times P(R) > P(D|NR) \times P(NR) \quad (5)$$

The likelihood ratio P(D—R)/P(D—NR) as a score to determine the likelihood of the document with representation D belonging to the relevant set.

Term independence or Nave Bayes assumption is used to estimate P(D—R) using computation of P(ti—R) for term ti. Likelihood ratio P(D—R)/P(D—NR) of documents are used as a proxy for ranking based on the assumption that highly ranked documents will have a high likelihood of belonging to the relevant set. [1]

With some reasonable assumptions and estimates about the probabilistic model along with extensions for incorporating query term weights and document term weights in the model, a probabilistic ranking algorithm called BM25 (Best Match 25) is quite popular. This weighting scheme has evolved from several versions of Okapi system. This method has been used successfully in several TREC evaluations. It has been shown in these competitions that Okapi variations are very effective for short query based document retrieval. The Okapi weight for Document dj and query q is computed by the formula below. Additional notations are as follows:

t_i is a term

f_{ij} is the raw frequency count of term t_i in document d_j

f_{iq} is the raw frequency count of term t_i in query q

N is the total number of documents in the collection

[1] Readers should refer to Croft et al. (2009) Pp. 246-247 for a detailed description

df_i is the number of documents that contain the term t_i

dl_j is the document length (in bytes) of d_j

avdl is the average document length of the collection

The Okapi relevance score of a document d_j for a query q is given by the equation below, where k_1 (between 1.0-2.0), b (usually 0.75) and k_2 (between 1-1000) are parameters:

$$okapi(d_j, q) = \sum_{t_i \in q, d_j} \ln \frac{N - df_i + 0.5}{df_i + 0.5} \times \frac{(k_1 + 1) f_{ij}}{k_1} \{1 - b + b \frac{dl_j}{avdl}\} + f_{ij} \times \frac{(k_2 + 1) f_{iq}}{k_2 + f_{iq}} \tag{6}$$

5.3.4 Semantic Model

However sophisticated the above models become, they can miss many relevant documents because those models do not capture the complete meaning or information need conveyed by the user's query. The process of matching documents to a given query based on concept level and semantic matching instead of index term matching is the basis for semantic models. This allows retrieval of documents even when they share meaningful associations not inherently observed (or statistically captured) with other documents that are relevant to the given query.

Semantic approaches have been developed to address the lack of knowledge based IR methods. These methods include different levels of analysis such as morphological, syntactic and semantic analysis to retrieve documents more effectively. In morphological analysis, roots and affixes are analyzed to determine the parts of speech (nouns, verbs, adjectives etc.) of the words. Following morphological analysis, syntactic analysis follows to parse and analyze complete phrases in documents. Finally, the semantic methods have to resolve word ambiguities and/or generate relevant synonyms based on the semantic relationships between levels of structural entities in documents (words, paragraphs, pages or entire documents). The development of a sophisticated

semantic system requires complex knowledge bases of semantic information as well as retrieval heuristics. These systems often require techniques from artificial intelligence and expert systems. Knowledge bases like Cyc and WordNet have been developed for use in Knowledge-based IR systems based on Semantic Models. The Cyc knowledge base, for example, is a representation of a vast quantity of commonsense knowledge about assertions (over 2.5 million facts and rules) interrelating more that 155,000 concepts for reasoning about the objects and events of everyday life. Wordnet is an extensive thesaurus (over 115,000 concepts) that is very popular and is used by many systems and is under continuous development.

5.4 Types of Queries in IR Systems

Different keywords are associated with the document set during the process of indexing. These keywords generally consist of words, phrases and other characterizations of documents such as date created, author names etc. They are used by an IR system to build an inverted index consulted during the search. The queries formulated by users should be comparable against the set of index keywords. Most IR systems also allow Boolean and other operators that can be used to build a complex query. The query language with these operators enriches the expressiveness of a users information need. While some retrieval models provide direct support for certain query types, some other types of queries can possibly be supported with pre-processing and workarounds. Note that we are not dealing with multi-lingual queries or queries against images and different formats of documents in this Chapter.

5.4.1 Keyword queries

Keyword-based queries are simplest forms of queries where the user enters keyword combinations to retrieve documents. This type of query has become the most commonly used in IR systems.

The keyword terms of the queries are connected with each other with an implicit

logical ‘AND’ operator. A query such as ‘database concepts’ retrieves documents that contain the words ‘database’ and ‘concepts’ together at the top of the retrieved results. In addition, most systems also retrieve documents that contain only ‘database’ or only ‘concepts’ in their text. Some systems remove most commonly occurring words (called stopwords) as a pre-processing step before sending the filtered keywords to IR engine. Most IR systems do not pay attention to ordering of these words. All retrieval models provide support for keyword queries.

5.4.2 Boolean queries

Some IR systems allow using “AND, OR, NOT, ( ), + , -” Boolean operators in combinations of keyword formulations. ‘AND’ requires that both terms be found. ‘OR’ lets either term be found. ‘NOT’ means any record containing the second term will be excluded. ‘( )’ means the Boolean operators can be nested using parentheses. ‘+ is equivalent to AND, requiring the term; the + should be placed directly in front of the search term. ‘- is equivalent to AND NOT and means to exclude the term; the - should be placed directly in front of the search term. Complex Boolean queries can be built out of these operators and their combinations and are evaluated according to the classical rules of Boolean algebra. No ranking is possible, because a document either satisfies such a query (is “relevant”) or does not satisfy it (is “non-relevant”). A document is retrieved for a Boolean query if the query is logically true as an exact match in the document. Users generally do not use combinations of these complex Boolean operators and IR systems support a restricted version of these set operators. Boolean retrieval model can directly support different Boolean operator implementations for these kinds of queries.

5.4.3 Phrase queries

When documents are represented using an inverted keyword index for searching, the relative order of the terms in the document is lost. In order to perform exact

phrase retrieval, these phrases should be encoded in the inverted index or implemented differently (with relative positions of word occurrences in documents). A phrase query consists of a sequence of words that makes up a phrase. The phrase is generally enclosed within double quotes. Each retrieved document must contain at least one instance of the exact phrase. Phrase searching is a more restricted and specific version of proximity searching that we mention below. A phrase searching query would be : "conceptual database design". If phrases are indexed by the retrieval model, any retrieval model can be used for these query types. Phrase thesaurus may also be used in semantic models for fast dictionary searching for phrases.

5.4.4 Proximity queries

Proximity search refers to a search that accounts for how close within a record multiple terms should be to each other. The most commonly used proximity search option is a phrase search that requires terms to be in the exact order. Other proximity operators can specify how close terms should be to each other. Some will also specify the order of the search terms. Each search engine can define them differently and use various operator names such as NEAR, ADJ(acent), or AFTER. In some cases, a sequence of single words is given, together with a maximum allowed distance between them. Vector space models that also maintain information about positions and offsets, tokens (words) have robust implementations for this query type. However, providing support for complex proximity operators becomes computationally expensive, suitable for smaller text collections in comparison to the Web.

5.4.5 Wildcard queries

Wildcard searching is generally meant to support regular expressions and pattern matching based searching in text. In IR systems, certain kinds of wildcard search support may be implemented; usually words with any trailing characters (e.g., data* would retrieve data, database, datapoint, etc.). Providing support for wildcard

searches in IR systems involves some overhead and is not considered worth the cost by many Web search engines today. Retrieval models do not directly provide support for this query type.

5.4.6 Natural Language queries

There are a few natural language search engines that aim to understand the structure and meaning of queries written in natural language text, generally as a question or narrative. This is an active area of research that employs techniques like shallow semantic parsing of text, or query reformulations based on natural language understanding. The system tries to formulate answers for such queries from retrieved results. Some search systems are starting to provide natural language interfaces to provide answers to specific types of questions, e.g., definition and factoid questions, which ask for definitions of technical terms or common facts that can be retrieved from specialized databases. Such questions are usually easier to answer because there are strong linguistic patterns giving clues to kinds of sentences, e.g., defined as, refers to, etc. Semantic models are used for providing support for this query type.

5.5 Text pre-processing

In this section we review the commonly used text pre-processing techniques (refer to the Text-processing task in Figure 34) that make the unstructured text in a document more amenable and efficient for information retrieval.

5.5.1 Stopword removal

Stopwords are very commonly used words in a language that play a major role in the formation of a sentence, but seldom contribute to the meaning of that sentence. Words that are expected to occur in 80% or more of the documents in a collection are typically referred to as stopwords, and are rendered potentially useless. Because of the commonness and function of these words, they do not contribute much to the

relevance of a document for a query. Examples include words such as: the, of, to, a, and, in, said, for, that, was, on, he, is, with, at, by, it. These words are presented here with decreasing frequency from a large corpus called AP89. The top six of these words accounted for 20% of all words, and the most frequent 50 words accounted for 40% of all text.

Removal of stopwords from a document must be performed before indexing and storage. Articles, prepositions, conjunctions and some pronouns usually may be classified as stopwords. Queries must also be pre-processed for stopword removal before the actual retrieval process. Removal of stopwords results in elimination of possible spurious indices, thereby solely compressing the size of index structure by generally about 40% or more. However, doing so could reduce recall if the stopword is an integral part of a query. (e.g.,: To be or not to be where removal of stopwords makes the query inappropriate). Many search engines do not employ stopword removal for this reason.

5.5.2 Stemming

A stem of a word is defined as the word obtained after trimming the suffix and prefix of an original word. For example, comput is the stem word for computer, computing, computation, etc. These suffixes and prefixes are very common in the English language for supporting the notion of verbs, tenses and plural forms. Stemming reduces the different forms of the word formed by inflection (due to plurals or tenses) and derivation to a common stem.

Stemming is the process of reducing a word to its stem word by applying any stemming algorithm. In English, the most famous stemming algorithm is the Martin Porter's stemming algorithm. The Porter stemmer is a simplified version of Lovins technique that uses a reduced set of about 60 rules (from 260 suffix patterns in Lovins technique) and organizes them into sets, with conflicts within one subset of rules

resolved before going on to the next. Using stemming for pre-processing data results in a decrease in the size of the indexing structure, and increase in recall, possibly at the cost of precision.

5.5.3 Thesaurus

A thesaurus comprises a pre-compiled list of important words in a given domain of knowledge, and for each word in this list, a set of synonyms and related words . This pre-processing step assists in providing a standard vocabulary for indexing and searching. Usage of thesaurus, also known as a collection of synonyms, has a substantial impact on the recall of information systems.

UMLS is a large biomedical thesaurus of millions of concepts (called meta-thesaurus) and a semantic network of meta concepts and relationships that organize the meta-thesaurus. The concepts are assigned labels from the semantic network. This thesaurus of concepts contain synonyms of medical terms, hierarchies of broader and narrower terms, etc that make it a very extensive resource for information retrieval of documents in the medical domain.

WordNet is a manually constructed thesaurus that groups words into strict synonym sets called synsets. These synsets are divided into nouns, verbs, adjectives and adverb categories. Within each category, these synsets are linked together by appropriate relationships such as class/subclass or is-a relations for nouns.

It is based on the idea of using a controlled vocabulary for indexing, thereby eliminating redundancies. It is also useful in providing assistance to users with locating terms for proper query formulation.

5.5.4 Other Digits, Hyphens, Punctuation marks, Case of letters

Digits, dates, emails, urls may or may not be removed from the text during pre-processing. Web Search Engines, however, index them instead to make use of this type of information in the document meta-data to improve precision.

Hyphens and punctuation marks may be handled in different ways. Either the entire phrase with the hyphens/punctuation marks may be used, or they may be eliminated. In some systems, the character representing the hyphen/punctuation mark may be removed, or may be replaced with space. Different information retrieval systems follow different rules of processing. Handling hyphens automatically can be complex: it can either be done as a classification problem, or more commonly by some heuristic rules. Most information retrieval systems perform case insensitive search. Thus all the letters of the text document are either converted to upper case (or lower case). It is also worth noting that many of these text preprocessing steps are language specific such as involving accents and diacritics and the idiosyncrasies that come with the language.

5.5.5 Indexing

Searching for occurrences of query terms in text collections can be performed by sequentially scanning the text. This kind of online searching is only appropriate when text collections are small. Most Information Retrieval systems process the text collections and operate upon the inverted index data structure. An inverted index structure comprises vocabulary and document information. Vocabulary is a set of distinct query terms in the document set and each element of this set has an associated collection of information about the documents such as document id, offsets within document etc. The simplest form of vocabulary terms consist of words or individual tokens of the documents. In some cases, these vocabulary terms consist of phrases, n-grams, entities, links, names, dates or manually assigned descriptor terms from documents and/or Webpages. For each term in the vocabulary, the corresponding document IDs, occurrence location of the term in each document, number of occurrences of the term in each document, and other relevant information may be stored in the document information section.

Weights are assigned to document terms to represent an estimate of the usefulness of the given term as a descriptor for distinguishing the given document from other documents in the same collection. A term may be a better descriptor of one document than of another by the weighting process.

5.6 Trends in IR

In this section we review a few concepts that are being considered in more recent research work in information retrieval.

5.6.1 Faceted Search

Faceted Search is a technique that allows for integrated search and navigation experience by allowing users to explore by filtering available information. This search technique is used often in ecommerce Websites and applications enabling users to navigate a multi-dimensional information space. Facets are generally used for handling three or more dimensions of classification. When, for the purposes of the classification, it is possible to organize the entities by three or more mutually exclusive and jointly exhaustive categories, facets are an appropriate classification scheme. A faceted classification system allows the assignment of multiple classifications to an object, enabling the classifications to be ordered in multiple ways, rather than in a single, pre-determined, taxonomic order. Unlike traditional category assignments, faceted search systems have documents existing in multiple overlapping taxonomies. A facet comprises "clearly defined, mutually exclusive, and collectively exhaustive aspects, properties or characteristics of a class or specific subject". For example, a collection of books might be classified using an author facet, a subject facet, a date facet, a country facet etc. Faceted search uses faceted classification that enables a user to navigate information along multiple paths corresponding to different orderings of the facets. This contrasts with traditional taxonomies in which the hierarchy of categories is fixed and unchanging. University of California, Berkeleys Flamenco

project is one of the earlier examples of a faceted search system.

5.6.2 Social Search

Web search has changed dramatically the way we interact with the knowledge of the world. Its success in impacting our everyday lives in the last two decades is perhaps unparalleled. Surprisingly, however, researchers have mostly thought about navigating and browsing for information as a single user activity, centered on eliciting users information needs and improving the relevance of search results as far as that user is concerned. This view is somewhat at odds with prior research by library scientists looking at users information seeking habits. This research demonstrated that other individuals may be valuable information resources during information search. More recently, researchers have observed direct user cooperation during Web-based information seeking. Some studies have reported that significant segments of the user population are engaged in explicit collaboration on joint search tasks on the Web. Certainly, active collaboration by multiple parties does occur under some circumstance (e.g., enterprise settings); at other times, and perhaps for a greater majority of searches, users may interact with others remotely, asynchronously, and even involuntarily and implicitly.

Socially enabled online information search (social search) is a new phenomenon facilitated by recent Web technologies. Collaborative social search involves different ways for active involvement in search related activities such as co-located search, remote collaboration on search tasks, use of social network for search, use of expertise networks, involving social data mining or collective intelligence to improve the search process and even social interactions to facilitate information seeking and sense making. This social search activity may be done synchronously, asynchronously, co-located or in remote shared workspaces. Social psychologists have experimentally validated that the act of social discussions has facilitated cognitive performance. People in

social groups can provide solutions (answers to questions), pointers to databases or to other people (meta-knowledge), validation and legitimation of ideas, and can serve as memory aids and help with problem reformulation. Guided participation is a process in which people co-construct knowledge in concert with peers in their community. Information seeking is mostly a solitary activity on the Web today. Some recent work on collaborative search reports several interesting findings and the potential of this technology for better information access.

5.7 Related Work

Powerset is a natural language search engine with semantic indexing, based on the XLE, Natural Language Processing technology licensed from the Palo Alto Research Center (PARC). During both indexing and querying, deep natural language analysis methods are used to extract semantic relations and semantic connections between words and concepts. Advanced search-engineering technology makes these facts available at query time for retrieval by matching them against facts or partial facts extracted from the query [37].

Another related problem of determining quality of queries is addressed in [38]. Various efforts [26] [110] have been made towards formalizing and understanding this problem. In the web environment, studies have shown that most users still enter only one or two queries, and conduct limited query reformulation [106] therefore automated query reformulations may help users in satisfying their information need better.

Some studies have also shown that users prefer the natural language enabled navigation in contrast to other ways of navigation such as menu-driven navigation [70]. In addition, the study confirmed the efficiency of using natural language dialog in terms of the number of clicks and the amount of time required to obtain the relevant information. In fact, user's interest in a website decreases exponentially with increasing number of clicks [59].

5.8 IR in Cobot

Information Retrieval in Cobot addresses the following issues from the perspective of information retrieval:

- Ability to retrieve relevant results from long interactive conversations
- Retrieval and Indexing of relevant information from web search
- Retrieval and Indexing of conversations.
- Retrieval, indexing and update of users conversations (for user search)
- Semantic indexing of above entities for beyond keyword retrieval
- Real time indexing and retrieval support for live performance

5.8.1 Ability to retrieve relevant results from long interactive conversations

A natural language search engine doesn't understand natural language; it incorporates information extraction techniques on text snippets to recognize the sentence focus, important entities in sentence, the relationship between entities and uses this information to formulate queries on the search index and provide relevant results. In fact, it comes very close in design to a question answering system whose additional tasks, beyond search, is to extract potential answers and select the best one from the retrieved results.

We use a combination of methods in Cobot to generate conversational queries. Cobot extracts domain-specific concepts, parses the text to extract associations between different phrases and uses these as candidates to generate multiple queries for the conversation in context. Cobot classifies each sentence to get the intent in the sentence and only uses sentences that get classified as questions, disclosures, edifications or advice to generate queries. Cobot drops sentences with acknowledgements,

confirmations, reflections and interpretation intent in conversations. We describe the methods for concept extraction and semantic parsing in the chapter on Information Extraction. Cobot also uses conversation memes, a data structure that stores the relevant extractions and generates queries from responses in conversations only if there is some new information generated in memes from responses.

5.8.2 Retrieval and Indexing of relevant information from web search

Cobot generates queries and sends them to Microsoft Bing search engine (unlike Google search engine, Bing does not have per day limit restrictions for search requests) as HTTP requests. Cobot also restricts searches to a list of domain specific or informational sites to avoid fetching irrelevant results due to bad query formation. Currently, Cobot restricts searches to sites like Wikipedia, Youtube, Yahoo answers, popular Open Education Resources (OER) resources, health sites like WebMD, PatientsLikeMe for it's searches.

5.8.3 Retrieval and Indexing of conversations

Cobot indexes conversations as they happen in the system so that they become immediately available for retrieval in the next cycle. Cobot shows related conversations for an existing conversation by searching over this conversation index and running the filters in later stages for final results.

5.8.4 Retrieval, indexing and update of users

When cobot suggests users for an existing conversation, it runs the user modeling based recommendation engine for computing candidate users. However, running the user modeling engine on each user in the community is not an option because of the real time nature of the problem. Therefore, cobot maintains a user index where it keeps indexing and updating users' conversations. Cobot retrieves a small candidate pool of potential responders by searching this user index. One potential problem we

face in this scheme is that the search index tends to retrieve users who are most active in the community for popular and common topics - this prevents new users from being retrieved for further upstream recommendation generation. We can partially address this problem by mixing this candidate pool with random selection of few users with scores above a minimum threshold.

5.8.5 Semantic indexing beyond keyword retrieval

Cobot uses additional information when available to add to the search index. For example, instances of diseases such as cancers, pancreatitis, etc also get indexed by the semantic types. Our medical dictionaries of over a million concepts have this associated information. This helps in retrieval of closely related items that do not match by keywords.

5.8.6 Performance

We measure how effectively Zoie system, the real time indexing and retrieval module, can perform real time indexing and search. We perform stress testing of Zoie engines that are integrated in Cobot infrastructure replacing the database based search.

Figure 35 measures the Transactions per minute that Zoie can handle. We shot continuos file indexing requests over a Memory Stream channel to Zoie and performed searches twice every second and ran this setup for 5 minutes continuously to measure performance. We performed this tests on a Macbook machine with 2 GHz Intel Core 2 Duo processor and 2 GB RAM. We define transactions in our scenario as a completed indexing and search request.

In Figure 36, we report the Average Transaction time of the system.

While performing these stress tests, we also monitored the load of the web server on which we had deployed these test requests. Figure 37 shows the overall memory and CPU load, number of running threads and the number of classes loaded while the tests ran. No other significant process was running on the system while these tests

Figure 35: Transactions per minute

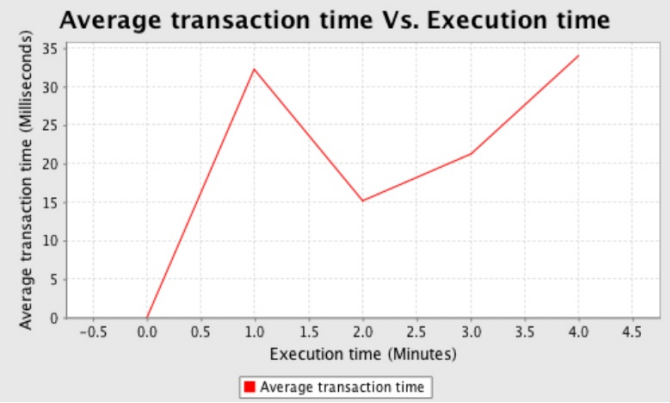

Figure 36: Average Transaction Time

ran. We conclude that Zoie performs very well in our setting and is capable of real time indexing and search requirement that we needed.

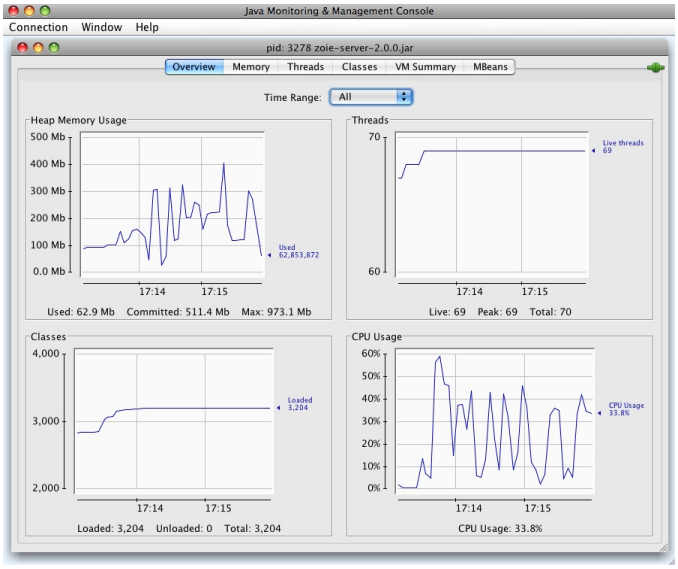

Figure 37: Web Server Load Monitoring

CHAPTER VI

SPEECH ACT ANALYSIS

6.1 Introduction

Virtual communities have emerged due to the recent advances in computer-mediated communication infrastructures and web technologies. Nowadays, people are always connected, whether in their daily lives or in their activities online. At the same time, the highly networked environment has trigged researchers to study how individuals behave differently in on-line social environments versus in more traditional face-to-face contexts. Online social communities are found to exhibit a more uneven participation distributions. Within small group sessions, it is common for the top few active participants to account for 50-75% of the communication activity, while the less active participants contribute very little relatively. Some early research shows that participation differentials may be due to status differences ([97], [114]) and differences in individual's expectations regarding participation ([92]). Research studies have looked into different ways to motivate community contributions[66]. use a theory of effectively managing group resources as design principles to analyze the successes and failures of Usenet. A significant amount of research has devoted to enhance online communities through expertise finders systems ([69], [64]). Such systems identify people who have expertise to answer certain types of questions.

Intelligent information agents in the community push questions to relevant users for answering and encourage users to ask questions. In order to effectively target users for question answering, we studied existing users and conversations of a medical community and analyzed their conversational patterns to create user specific recommendation models. In the approach we have described, the recommendation model

depend on user's intentions while adhering to the overall community based intensions.

6.2 System Description

Verbal Response Modes (VRM) is a principled taxonomy of speech acts that can be used to classify literal and pragmatic meaning within utterances. [71]. In this classification, utterances are classified into disjoint sets comprising Question(Q), Disclosure(D), Edification(D), Advisement(A), Acknowledgement(K), Reflection(R), Interpretation(I) and Confirmation(C). We crawled 12000 conversations from WebMD forums consisting of 3260 users to train and test our VRM classifier.

Choice of Features The choice of features to predict the type of utterances is extremely important. We have used a mix of contextual, syntactic and semantic features for our data. We have extracted the following features for our task: *Number of words*, *First word*, *Last word*, *bigrams*, *Dependencies (1st/2nd/3rd person subject, inverted subject-verb order and imperative verbs.)*, *Morphology*, *Hand constructed word lists*, *Wh words*, *Top n words*.

In order to develop the VRM classifier that could categorize the conversations at sentential level, we (two annotators) manually tagged 175 conversational sentences for a total of 1941 instance training set including the VRM training data. We report our 10-fold cross validation accuracies as follows. Our SVM based classifier achieves 10 fold cross-validation precision of 72.3%, recall of 75.3% and a F-Measure of 73%. In our classification task, we combined Disclosure and Edification classes into one as our classifier was confusing with these categories and we didn't need to model these as separate categories. Our top features in this classification task were domain independent features such as '?', length of words, 'you', 'i','okay', 'well', etc.

6.3 Community Modeling

After training, we ran the classifier on the crawled data and analyzed the top 100 most active users in the forum. Our goal is to create three different users models, an

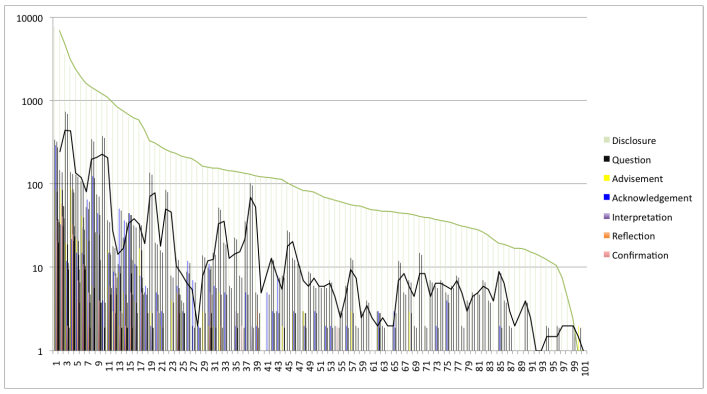

Figure 38: Community Intentional Analysis

enthusiasts model of active participants, a casual user model and a passive user model so that we can target users in these groups differently in order to engage them into the conversations more effectively. Our other goal is to not inundate the enthusiasts with very frequent requests for recommendations thereby annoying them with the system.

Our observations are summarized in the Figure 38. The first quartile (42.5 users) consists of very active users (note the Y-axis logarithmic scale), the next quartile consists of casual users and the third and last quartiles consist of passive users community. Surprisingly, the most active user in our snapshot, had only asked 5 questions but had contributed in many different conversations. Not so surprisingly, many of the top users were WebMD forum personnel who actively sought to increase participation in the community. We also noticed that people in casual and passive user model groups asked more questions compared to the enthusiasts in this context as shown in Figure 39.

We propose the following approach for the community based recommendation model that takes into account user engagement levels while making recommendations:

1. Classify users' responses into questions (Q) (Question VRM category), answers (A) (Disclosure, Edification and Advisement VRM category) and miscellaneous (O) (Acknowledgement, Interpretation and Reflection VRM categories)

2. Learn a question-answers-miscelleneous proportion of Q:A:O of enthusiasts, casual and long

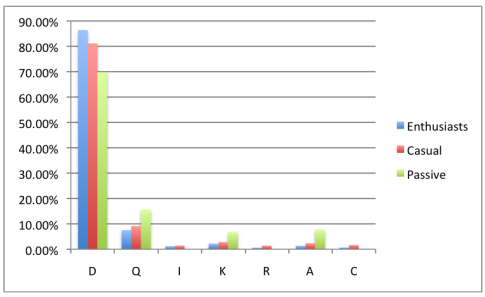

Figure 39: Community Intentional Parameters

tail user models depending on community engagement (with A more than Q more than O)

3. Categorize each user into the enthusiast, casual or passive user model.
4. Given a question, calculate top n Q-list, A-list, O-list recommendations based on topic relevance model
5. Re-rank topic relevance model based on User's Q+A+O current engagement (if a user is under engaged, prefer him before others, don't choose this user if he has already met the demands)
6. Recommend users proportional to their group user model Q:A:O ratio
7. Update proportions after users have responded.

CHAPTER VII

USER MODELING

7.1 Introduction

Push-based information delivery systems are those where the information is supplied to the user rather than the user typing in her information need expressed via a query each time. Recommendation systems constitute such systems that make user aware of content satisfying their information need without the user having to ask for it each time. In such a setting, there is a huge opportunity to do the task of recommendation accurately by maintaining rich domain, user and interaction models in order to personalize content. This is a very hard problem that involves understanding user's dynamic information need, interests and what they might find worthy of their time worth at 'now' point of time. Various dynamic parameters such as user's short term interests, longer term interests, current information need and context, user's tolerance levels for exploration vs. exploitation of information, her existing knowledge levels, etc. need to be accounted for for this user modeling task. In fact, in many situations, users' themselves aren't aware of ways to explore the problem space, especially, in the context of e-Learning and knowledge navigation environment. Any fine-grained user modeling system should account for parameters such as novelty, diversity and similarity while making recommendations using the model.

The field of User Modeling involves continuously capturing, storing, selecting,inferring and predicting implicit information about users from explicit information about them. Models of implicit information attempt to encapsulate user's characteristics, behaviors

and preferences, differences between user's characteristics and expert model characteristics and/or the cognitive processes underlying actions of the user. Modeling systems/agents have to make several assumptions about users in the absence of complete information to perform acquisition, representation, learning and reasoning tasks. The models can be application or domain specific, structured or semi-structured, symbolic or semantic. User Modeling still poses as a challenging topic with not much breakthrough despite several advances in machine learning for user modeling technologies. The critical issues that limit the real world application of user modeling may be attributed to: [112]

1. the need for large data sets;
2. the need for labeled data;
3. concept drift; and
4. computational complexity;

Using user modeling systems, for example, personalized learning resource can be generated for a learning community to match learner's individual preferences and levels. Common interest cohorts can be formed and learners can be efficiently routed to other learners for reaping the benefits of social and peer learning.

7.2 Related Work

User modeling has been researched by a number of researches in applications like interactive tutors, user action predictors and product recommendation systems. [12] have used user modeling for a tutoring system where they try to model a student's understanding of a particular concept. Similar work was done earlier by [25], [23] and [27] who used user modeling in the form of overlay models. [29] have explored machine learning based classifiers like Decision Trees for prediction of user actions while [8] have used a naive bayesian classifier for the task of identifying interesting websites

for a user. In another research [16] explored the use of simple keyword matching based system for personalized tracking of scientific literature on the web. They have implemented this system as a part of the CiteSeer digital library system. Their goals though similar to ours, has traded richness of model for faster calculation. We believe that having a rich user model, though slower to some extent, is worth the cost for tackling the problem of information overlaod.

Capture of implicit information for dynamic and semantic user modeling involves techniques in Knowledge Representation (KR). Knowledge Representation has long been considered one of the principal elements of Artificial Intelligence, and a critical part of all problem solving. Many powerful meta models of semantic networks have been developed such as Existential Graphs [83] of Charles S Peirce, Conceptual Graphs [105] of John F Sowa and the Resource Description Framework [72] by the World Wide Web Consortium. These rich intermediate representations try to capture the natural language into logically precise and humanly readable forms amenable to semantically intelligent and extensible post processing for supporting access and reasoning tasks.

Researchers have explored linguistic and semantic knowledge representation techniques to address problems with traditional bag of words representation approach from IR field. [78] showed that syntax analysis on text can improve the retrieval performance. [19] used domain specific hand-coded thesaurus to improve the performance of retrieval. The work of [22] showed that inclusion of semantic information from sources like the WordNet [47] can considerably improve the performance of the bag of words technique. [91] describe a hybrid CBR-IR system in which CBR is used to drive IR. Their system uses a standard frame-based representation of a problem and matches it against frame-based representations of cases using a previously developed CBR system called HYPO. Documents stored in those cases are then used to construct keyword queries, which are run against a large document repository using

the INQUERY search engine. This approach relies on frame-based symbolic AI representations. Their approach returns a potentially large set of documents for the user to interpret.

Case Based Reasoning is another related technology related to user modeling and recommendations via representation of 'cases' or snippets of frame based evidential experiences for problem solving applications. CBR technology is being applied in areas like Question Answering [22], Knowledge Management [113], and Information Retrieval [91]. Traditional approaches in textual case based reasoning to indexing and retrieval are based on the Vector Space model where each case is represented as a feature vector (bag of words notation). The similarity measure is based on the cosine distance between the feature vectors. This approach, however suffers from problems like synonymy (different words have same meaning) and polysemy (same word has different meanings in different contexts).

Recently, researchers have used graph-based technique for such knowledge rich representations. [101] propose the usage of Semantic Graph Model (SGM), while [75] develop a semantic graph based on extracting triples using deep semantic analysis of text. The advantage of such graph base approach is that we can employ graph based algorithms on them and do interesting things. [10] propose the usage of Spreading Activation for IR, [101] propose the usage of graph structural matching for similarity calculation, while [60] use Google Page Rank [18] based approach for document summarization.

Soar (State, Operator and Result) [81] is a cognitive architecture that grew out of problem solving and learning research with the goal towards building autonomously interacting intelligent systems in complex environments. Soar models human cognition as a symbolic production based representation system [61] with memories (long term production memory, working memory and preference memory) and processes (input, output, decision procedure and learning). Tasks in Soar are carried out by a

search in problem space and new chunks are created with dependency analysis of the decision cycle while resolving impasses. An impasse models a situation where a user lacks adequate knowledge to complete a task.

Unlike Soar, ACT-R system [9] is developed using a hybrid model, with both symbolic and sub symbolic levels of processing. ACT-R operates as a rule based system storing and retrieving facts from declarative memory and using production rules for procedural memory. The sub-symbolic level models the variability of human behavior by implementing a semantic network for long term memory.

Other popular approach of user modeling in recent years is use of collaborative filtering technology for user/item recommendations. Explicit feedbacks from users ('collaborative') are extrapolated with statistical techniques to construct similarity matrices for user modeling and recommendations. Some personalization systems also capture feedback implicitly by modeling user behaviors such as clicks and mouse movements. GroupLens system [68] in its earliest flavors filtered Usenet news using collaborative filtering algorithm. One earlier hybrid recommendation system called Fab [13] used collaborative filtering with content analysis for recommending Webpages to users.

Adaptive information filtering systems that use explicit feedback to create and augment user profiles in a learning based setup have been widely studied in literature. [118] [24] [6]

7.3 User Modeling in Cobot

Cobot takes a hybrid learning approach to user modeling by learning from user's past behavior as an indicator for her future behavior. This past behavior includes watching conversations that users are participating in, pages and conversations they are clicking and the explicit ratings they are providing on such entities. User Modeling in Cobot is designed to help users get notified about related conversations, documents and people

at times they are being accessed in the community. User models are captured and learnt through content extraction from conversations, relevance feedback and click monitoring. Cobot maintains rich user models maintaining short term and long term profiles of the user. The process involves extraction and storage of domain concepts from user's conversations or other participating conversations and documents. Factors such as semantic similarity between concepts, recency of information, learning and unlearning of concepts, weights and their associations are modeled and used as filters for generating proactive right time user access user information. It is not just important to recommend right information to users but to recommend such information at 'now access' time or at times when there is some activity in the recommended information source . This helps in getting people engaged in conversations when it's happening in real time, thus providing access to such information at the right time. We feel that such a system would be very useful for e-learners and knowledge workers where the users can stay informed and connected on the latest community interactions on their topics of interests. Cobot does not aim to model user's understanding, but rather her potential knowledge and interests using concepts, ratings, semantic nets and some basic activity statistics. Such a system is inherently better suited for longer term learning tasks compared to other tasks such as single shot transactional web search and ad hoc retrieval. This modeling approach affords us to get direct user intervention as well, if needed, to update the users models. Many statistical machine learning based approaches do not provide this option to end users.

Figure 40 portrays our conceptual view of modeling user interactions with a short term model for capturing recent interaction episodes and a long term model for capturing the long term semantic net profile of the user. We will now briefly describe the different concepts and components needed to realize this model.

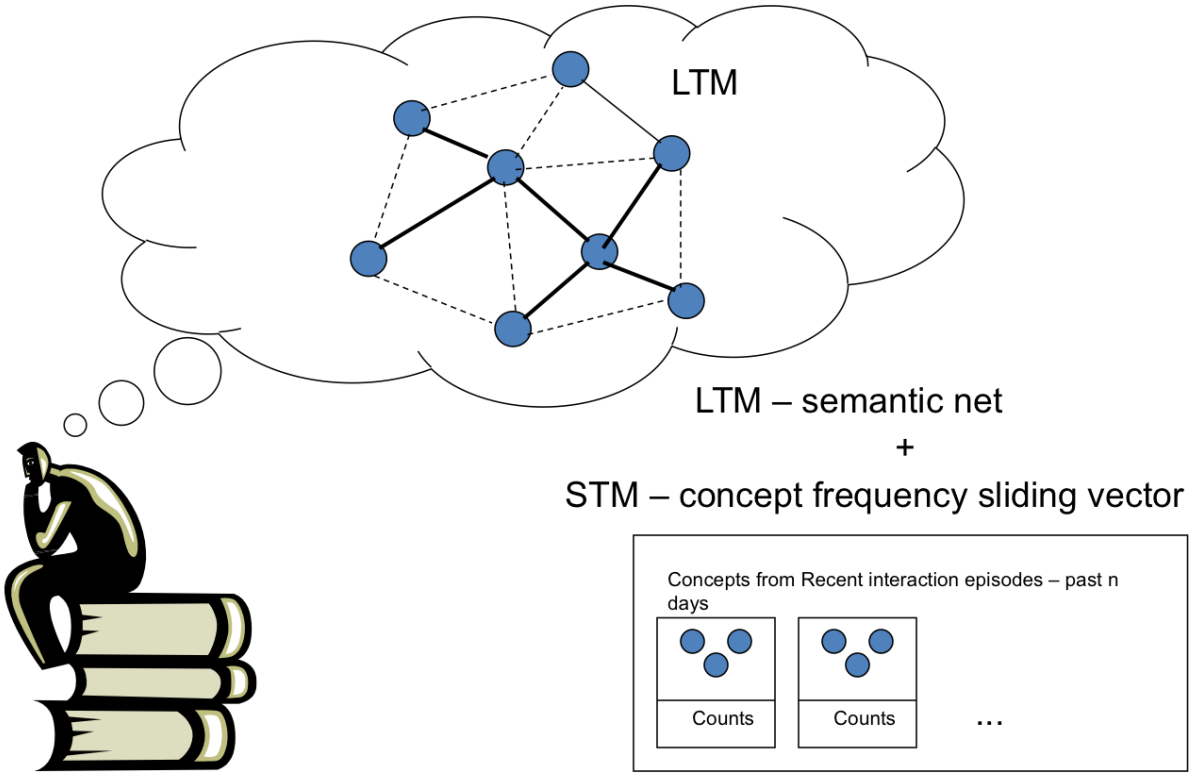

Figure 40: User Model

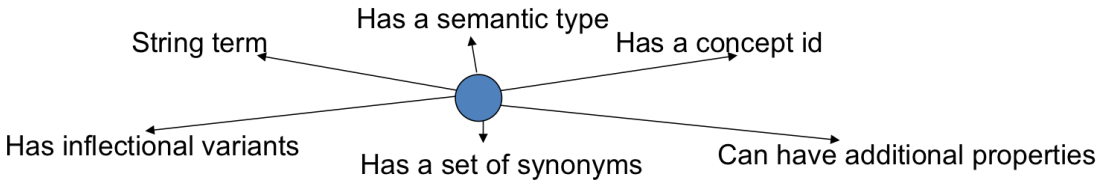

Figure 41: Concept Representationl

7.3.1 Concept

A concept 41 forms the core of our representation. Each concept represents a word along with associated information[reference to concept mapping system]. The models store as much information about concepts as desirable coming from upstream processes. Where possible, concepts have associated semantic types, parts of speech, their overall and short term frequency counts, etc. represented in the system.

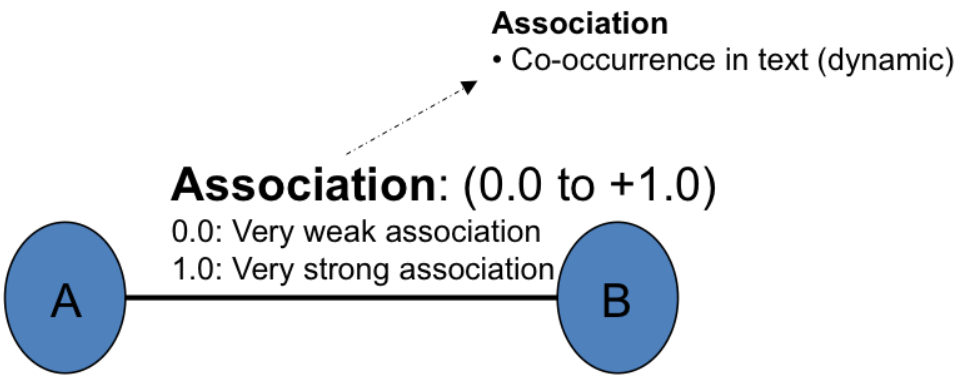

Figure 42: Assocation

7.3.2 Association

Concepts are connected using links that are called associations 42. The associations capture co-occurrence of concepts. If any two concepts appear often together then they would develop a strong association. The strengths of association is a weight between concepts that is determined by factors such as co-occurrences in conversations and documents, learning and unlearning rate parameters in the semantic net. By default there is no association between concepts.

7.3.3 Short Term Model

The purpose of the Short-term model (STM) is to capture the user's short-term interests which are concepts collected from recent user interaction episodes from conversations, feedbacks and page clicks. The STM is marked by a sliding window 43 which is the number of days in past from today whose events can be considered as 'current' for the user. We represent the STM as recent concept vector instances and use IB1 instance based learning function [5] to filter users matching the target conversation. Our similarity function uses semantic similarity functions [100] based on Wordnet hierarchies when available or falls back on edit distance based Smith-Waterman String similarity function when words are unidentified in Wordnet. The concept similarity scores are decayed exponentially using the following function 7:

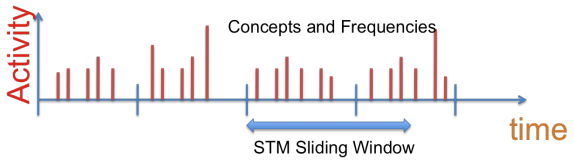

Figure 43: STM activity sliding window

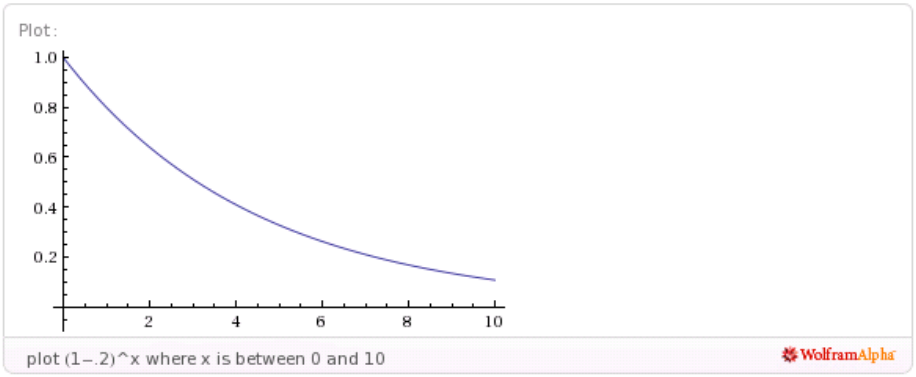

Figure 44: Per day Decay in STM scores

$$Score = Score \times (1 - 0.2)^x \tag{7}$$

In 7, x is the number of days since the last update.

The decay function is plotted as follows 44:

7.3.4 Crossover

Crossover 45 process takes instances from STM window and adds them to the LTM semantic net if the concepts to be added are above certain threshold frequencies. Currently all the qualifying concepts get added to the LTM, but a better approach would be to have some classifier decide if a particular instance should be discarded or added to the LTM. When a user engages in a conversation, rating or a document click episode, cobot determines if the time lapse between current interaction and last crossover operation stretches over the STM window size. If so, crossover operation is performed and last crossover operation times are updated for the user.

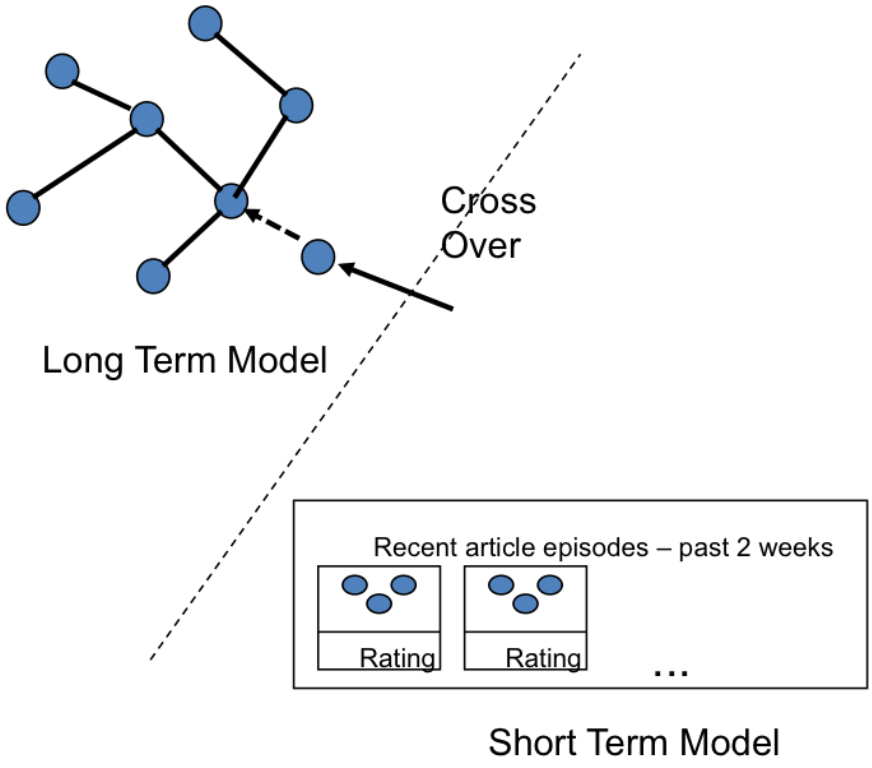

Figure 45: Crossover Operation

7.3.5 Long Term Model

The Long-term model (LTM) captures the user's long-term associated interests. This model tries to capture concepts and their co-occurrences that interest the user in general and for a prolonged period of time. We represent the LTM in the form of a Semantic Graph. The nodes of the graph are concepts the user is interested in. The concepts are connected with associations which develop when concepts co-occur frequently in user activities. Initially the LTM contains no concepts and starts building up after the first crossover operation. Over a period of time when the user engages in more conversational or interaction activity, new concepts are added to the LTM. All the concepts that appear in the LTM have a rating associated with them. The concept rating is computed using the following function(8):

$$Rating = 1 - \frac{1}{Concept_{freq}} \tag{8}$$

For a new rating obtained for an already existing concept in the LTM, the rating is adapted as follows (9):

$$Rating = Rating_{old} + LearningRate \times Rating_{new} \tag{9}$$

The Learning Rate parameter decides how much of the new rating to incorporate in the already existing rating. As mentioned earlier, the concepts are connected using associations in the LTM. Associations just capture co-occurrence of concepts in conversations and documents. When concepts belonging to a conversation, for example, are added to the LTM, they might strengthen old associations or create new ones. The strength of an existing association is updated using the following formula 10:

$$Association = Association_{old} \times (1 + LearningRate) \tag{10}$$

We have also implemented 'unlearning' or 'forgetting' of concepts from LTM that do not appear too often. The need for this was felt as many outlier concepts spring up in the LTM incidentally. So such concepts are slowly weakened and when they go below a certain threshold they are removed from the LTM permanently.

7.4 Recommendation

The social recommendation process involves scoring each user to select the top N users for conversational recommendation (Other filters are also applied in the user recommendation process, we will discuss those in other chapters). Figure 46 shows the user model based selection process.

The following steps are involved in the user model based recommendation process:

1. *Pool selection*: The user model based recommendation filter is an expensive filter. We need to select a small subset of users on which to run the recommendation engine. This is accomplished by querying our semantic search index for

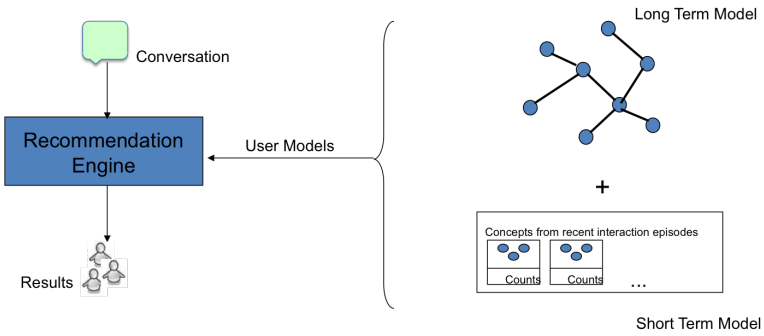

Figure 46: User Model based recommendation

users that match the queries in conversations.

2. *Concept Extraction*: This step is to extract concepts from conversations and documents in the manner discussed earlier. We use the downstream concept extraction system to collect these concepts.

3. *k-NN in STM*: This step involves finding the k Nearest Neighbor users matching the set of concepts in the conversations for which recommendations are being sought. This step basically tries to find a recent user whose STM matches the concepts from the current conversation. Doing the k-NN in the STM gives a measure of favorability (or otherwise) of the recommended user based on the recent events.

4. *Spread Activation in LTM*: This step involves performing spreading activation in the LTM network. For each of the concepts for the current user, a node 'fires' in the LTM and gets activated. The degree of activation is the similarity score of the new concept and the firing node. The activation is spread to the neighboring nodes proportional to the weight of each connecting association. The maximum radius of spread is a variable parameter; a larger spread radius impacts the performance of the overall recommendation engine therefore we use a spread radius of 3 for practical purposes. For all the nodes where activation spreads we sum the degree of activation with the score of the node. The output

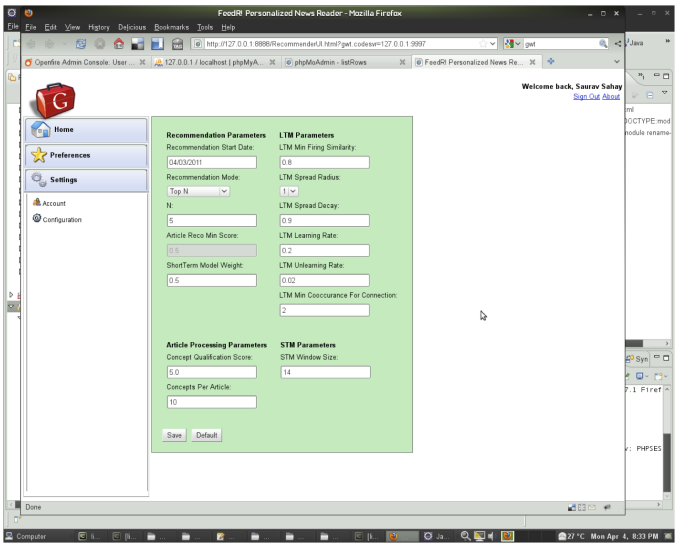

Figure 47: User Model

of this step is a weighted score of all the activated nodes.

5. *Score*: The net score of a user depends on its short-term and long-term score. The net score is calculated as follows11:

$$NetScore = STMScore \times STMWeight + LTMScore \times LTMWeight \quad (11)$$

6. *Rank*: Users are sorted based on the net score and the top N users are forwarded to the next filtering engine.

We have heuristically fixed the Short term model and long term model weights but these can be explicitly set by users. There are other parameters in the system as well that users can set for biasing the models to their preferences. Figure 47 shows the parameters that can be set by users in this user modeling engine. We feel that these parameters in a sense define the user's inherent preference. If the user has some long-term information goals (learners, doctors, patients, programmers etc) then the LTM weights needs to be more and if the user is just looking for the latest current recommendations (a casual user) then STM weights should be more. Currently though these parameters are set manually, we feel that they can be inferred as well by analyzing the user's feedback patterns.

CHAPTER VIII

SOCIAL FILTERING

With the advent of open education resources, social networking technologies and new pedagogies for online and blended learning, we are in the early stages of a significant disruption in current models of education. The disruption is fueled by a staggering growth in demand. Open Social Learning systems open a new venue for self-motivated learners to access high quality learning materials. These open learning communities are made up of users who are grouped by different information needs into dynamic cohorts. These social community dynamic networks, through effective sharing and collaboration, increase the overall utility of their online communities and help to solve individual problems more effectively. Cobot tries to leverage from the online community interaction network to incentivize it's recommendations towards the natural social dynamics of the community.

8.1 Social Learning Community

The idea of social learning community grows out of the success of social communities powered by today's web 2.0 technology and social network services. Facebook and Twitter are two examples of the most well known social networking websites. On such websites, users are represented by individual profiles and are connected to other users for production and consumption of content. Users can establish links with each other, attend user groups, send and receive messages, update their own profiles or statuses, get automatic notification about other users' updates, comments or rank other users' behaviors and so on. Essentially, social networking services provide a means for users with shared interests to interact over the Internet on a variety of topics, ranging from personal lives to business. The users who are linked to each other both explicitly

and implicitly (through secondary links) form a social community. Online forums can also be viewed as an initial form of online social communities. For example, WebMD (http://www.webmd.com/) holds a medical information forum that allows people to ask health related questions and to get advice. While these forums provide basic services for people to communicate with each other, they often do not support social navigation and activities such as following another similar user's interests or establishing social links with others. As a result, these forums do not provide a social infrastructure, often vital for supporting and sustaining longitudinal participation and engagement in the community. Also, social engagement provides an effective means to assess and analyze a user's reputation, expertise and other social capital metrics.

Intelligent question answering systems also make use of online social communities. For example, Aardvark (http://vark.com) is a website that allows a user to ask a question and get answered by another user in the user's extended network (including a user's friends' friends) by analyzing user profiles and past activities. This service is convenient for someone who is looking for an opinion from a person instead of a search engine. While this service benefits from information derived from the social network, it does not foster an organic growth of the community by sharing and leveraging from the information and connections created by users over time. IM–an–Expert [90] is another instant messaging based question answering service that identifies experts with potential knowledge about asked questions and routes the questions to these experts. User profiles are created in this service either explicitly by users by specification of keywords of interests or implicitly by extracting keywords from emails sent by users to mailing lists. Experts are matched by performing vector based searches using TF-IDF scoring multiplied by a temporal decay function to discount new messages from old ones.

It is well established that learning occurs in social context. Social learning theories

describe how learning occurs in communities [111]. Social learning is the acquisition of knowledge that happens within a social group - the process by which individuals observe the behavior of others and modify their own behavior accordingly. It is noted that individuals learn best by observing others, and are tremendously influenced by the role models they observe. Communities succeed through social learning because they provide opportunities for its members to observe others, to pay attention to role models, and to be motivated by the group to succeed. Additional support for positive outcomes in learning communities also comes from the work on peer-assisted tutoring [51], which shows peer-tutors benefit as much from tutoring as their tutees because the tutors structure their own knowledge during tutoring.

8.2 Social Capital

Social capital in general refers to features of social organizations such as networks, norms, and social trust that facilitate coordination and cooperation for mutual benefit. Reciprocity is a key mechanism for explaining how social capital functions among individuals [14]. In the context of online social community, reciprocity means people benefit from the community and also give back to the community. In this sense, establishing social capital aligns with the goal of a social community, which is to serve its members while growing by members' supports. Moreover, social capital gained from online communities can also be transferred to offline contexts [80]. For example, Facebook is used by people to maintain weak social ties, such as staying in touch with acquaintances from high school, or to bond close ties, such as emotional support for family members. Although different types of social capital created from online communities are not equally convertible into economic, symbolic, or cultural capital offline, researchers suggest that positive social capital outcomes can include career advancement, better public health, and organizational success. We believe social capital gained from an online social learning community can be very valuable since this type

of social capital represents one's education, professional skills, and expertise. In fact, It has been shown that directed communication between individuals have increased bonding social capital of a user [21].

In a community question answering system, for example, there is generally not a notion of explicit friendship network. However, there are some recent CQA systems such as Quora.com that allow for explicit asymmetric connections in the community to receive push notifications from the community. Besides the explicit network, there is a social network built implicitly based on user user interactions in the community. We leverage from the works of [102] to compute this implicit and explicit affinity networks in the community.

$$bonding(i,j) = s_{ij}^{IAN} \times s_{ij}^{ESN} \tag{12}$$

$$bridging(i,j) = (1 - s_{ij}^{IAN}) \times s_{ij}^{ESN} \tag{13}$$

In Equations 12 and 13, s_{ij}^{IAN} is the score of the Implicit Affinity Network between users i and j, and s_{ij}^{ESN} is the score of the Explicit Affinity Network between users i and j. The implicit affinity formulas work on a set of attributes (here topics, groups, description, school), where each attribute has a set of possible values (for example, topics=math, health sciences, cs, physics, biology). The attribute affinity scores are the number of values in common divided by the total number of possible values, so for each attribute, the affinity is higher if the users have more values in common. The overall implicit affinity score is the sum of the affinity scores for each attribute divided by the number of attributes the two users both have values for. The explicit affinities are 1 if users have ever had a directed communication in the system before.

In Cobot, when a user asks a question or responds in a conversation, the bonding capitals between the participants in the conversation and the asker/responder is computed and used as a signal to boost the recommendations.

8.3 Feedback

Besides filtering based on user's bonding capital, cobot also monitors user clicks and explicit feedbacks and registers them at individual preference level, conversational level and the entity level. For example, if a document receives a positive rating in a conversation, the document rating increases (entity level), the user-document rating increases (individual level) and the recommended document rating increases (conversation level). This helps in promoting community preferred results as recommendations in the system. If a recommendation has received better than average rating in the system in past, cobot adds a small factor (normalized average rating) to the final score of the candidate recommendation. The implicit feedback through user clicks is not currently being used in score modifications but this is an important feature that will play a role in a recommendation system in live deployments.

CHAPTER IX

DESIGN AND PROTOTYPE

9.1 Characteristics of the Tasks Performed by Users

Information seeking is mostly a solitary activity on the web today. Socially enabled online information access is a new phenomenon partly facilitated by recent web technologies and success of popular social sites. This collaborative social access involves finding together specific resources and people that can help you with your task at hand in more productive ways. Our premise is that traditional search engines, content portals, and forums are not being very useful for knowledge access that results in learning and engagement. The question we try to seek answers for is how can people use the web and technology for a blended learning atmosphere that provides them resources for better and more meaningful information access solutions?

Figure 48 suggest some of the primary factors why people go online for health information seeking. We see that people most people use the internet for seeking information related to disease, symptoms, treatment, specific conditions, etc. One key finding reported [44] is that consumer generated content appeals to consumers in decision mode while making healthcare choices for services, available options and reputations. Also, search engines remain the gateway for online health information [44]. We clearly observe that these two activities with respect to decision making with the help of consumer generated content and information access using search engines require access to different systems and effort.

We looked at the workflow for accessing information and noted that there were few clicks and access efforts we could possibly save by having an integration of some of the search methods using cobot.

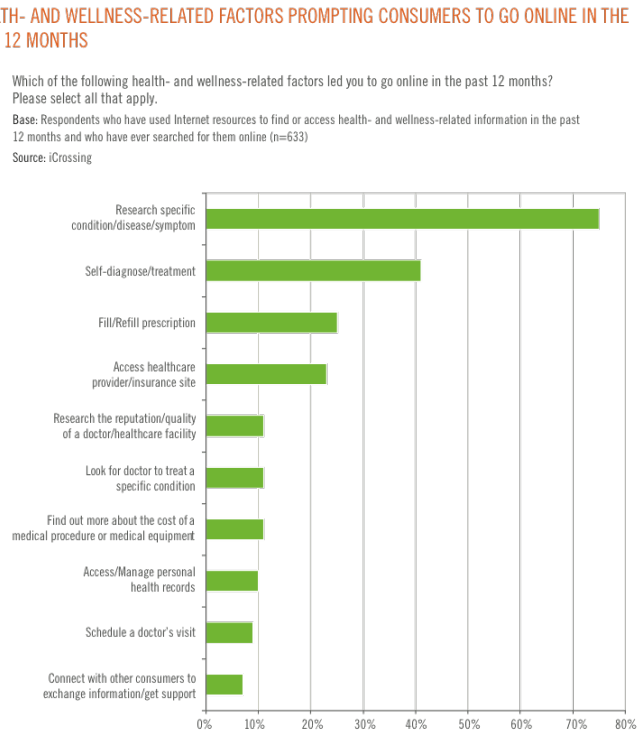

Figure 48: Why do health consumers go online? Source: N. Elkin, How America Searches: Health and Wellness, iCrossing Report; 2008

9.1.1 Hierarchical Task Analysis

This Hierarchical Task Analysis of Information Access using the web is outlined as follows:

1. Access Information
    (a) Access information on the web
        i. Use keyword-based search (informational search)
            A. Search keywords
            B. Scan pages for link
            C. Click link
            D. Scan pages for information
        ii. Use question-based search

A. Search questions

B. Scan pages for link

C. Click link

D. Scan pages for information

iii. Visit a Website that you already know that may have the information

A. Scan pages for link

B. Click link

C. Scan pages for information

iv. Post questions

A. Write a question and post

B. Wait for others to post comments

C. Comment on others answers

v. Have online conversation

A. . Initiate an online conversation

B. Discuss questions with participants

9.2 Design Choices

We came up with design alternatives for our integrated web information access system as shown in Figure 83.

We designed a system allowing people to initiate an online conversation, ask questions and offload the task of search to cobot agent and get in relevant recommendations straight into the conversation for all participants to benefit from new external knowledge and additive recommendation updates and interaction.

9.3 Widget based Prototype

We moved to another model for evaluating Cobot system as we realized that getting real users on our system was a task beyond the scope of this dissertation. We,

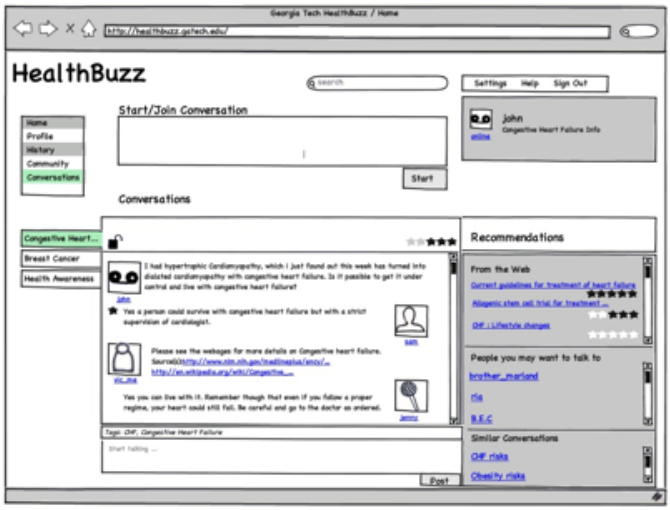

Figure 49: Rough Mockup design

therefore, partnered with an existing e-Learning website that was very close in concept to what we were looking for in Cobot system. This website, called Openstudy.com, allowed users to ask questions and do conversations in real time on different study related areas including the health and biomedical domain, as well as the sciences and arts domain. We developed a browser script that, once installed on the browser, could transfer conversations from this site to the cobot server, process them in realtime and send back recommendations. In the process, cobot indexed the conversations and recommendations along with capturing the user models for the users in different conversations.

Figure 51 shows the Openstudy.com landing webpage on which our browser based script was triggered.

Figure 52 shows the snapshot where a user clicked on a conversation and the conversation was sent to the cobot server. You see the message that says 'Analyzing question and fetching recommendations'. Cobot was able to pick up responses in real time or in later interaction episodes from the site and send them to the server for incremental updates.

Figure 53 shows some recommendations that were sent to the frontend display for

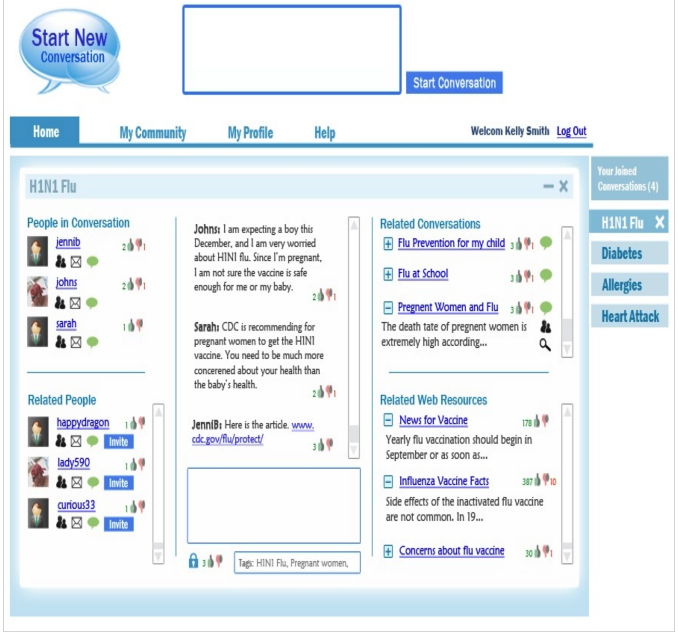

Figure 50: Cobot Interface

the conversation shown. Cobot captured both implicit and explicit feedback (through ratings) from the system.

Figure 54 shows push notification to a user who was recommended in another conversation. Whenever a user logged into the system, the cobot messaging server would fetch all notifications and display them to the user as shown here.

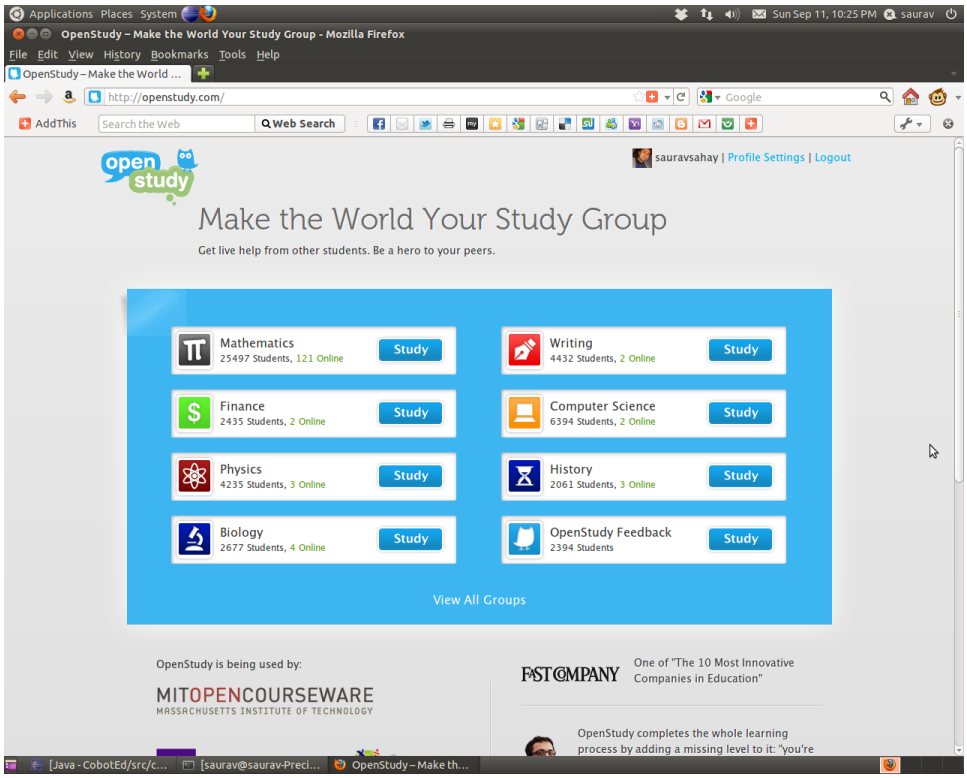

Figure 51: Browser plugin script for Openstudy - Login

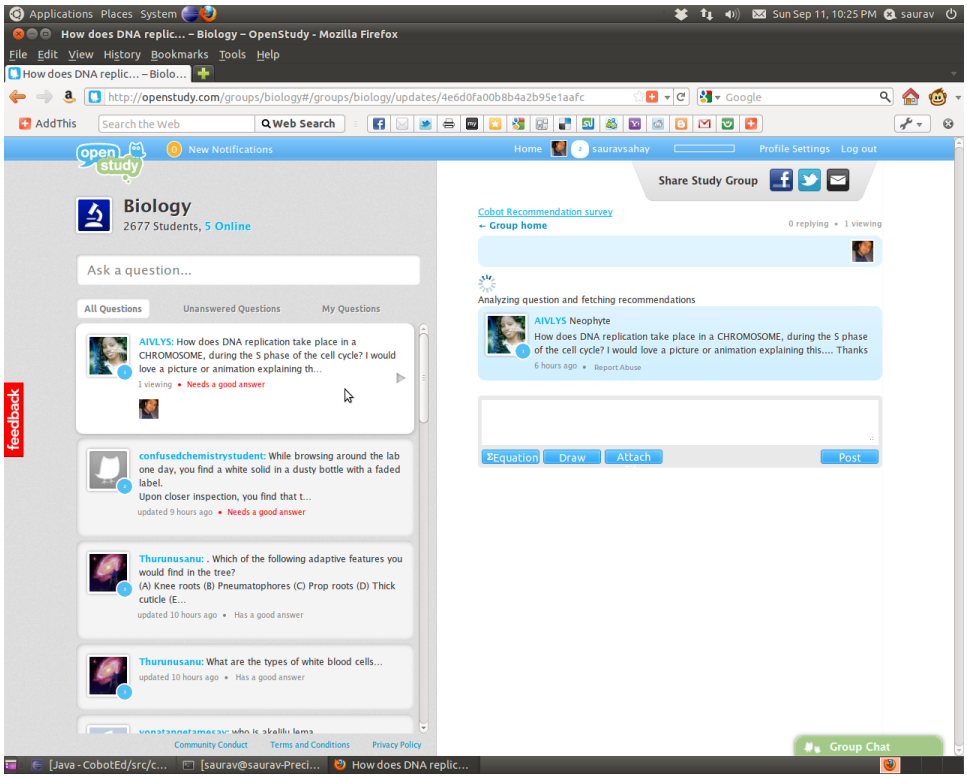

Figure 52: Browser plugin script for Openstudy - Analyzing conversation

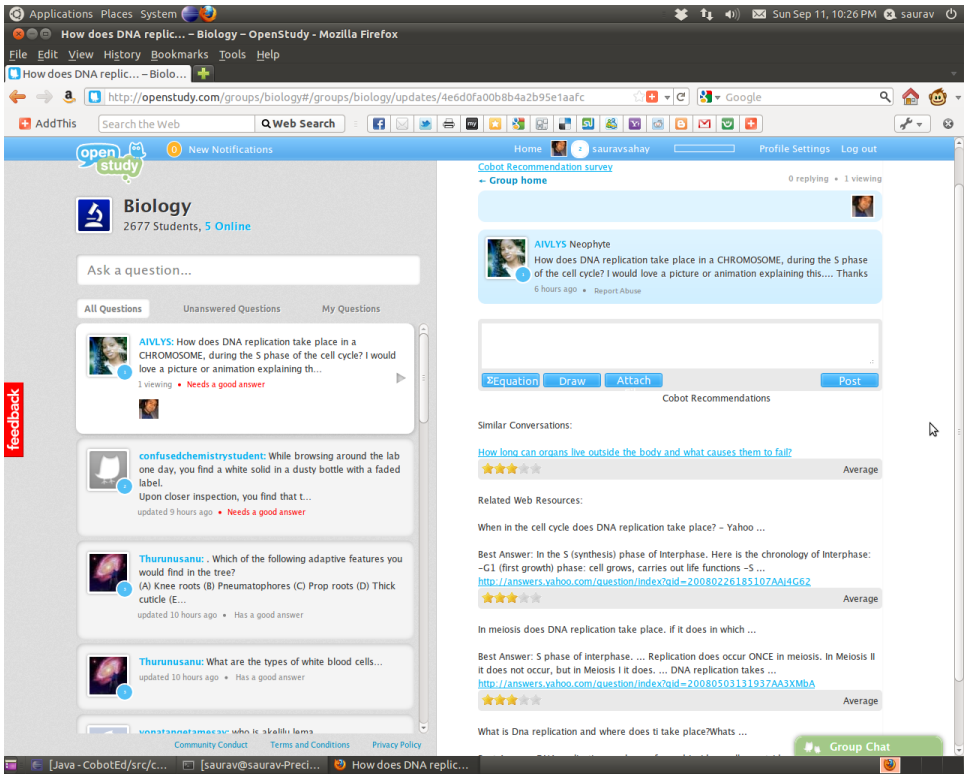

Figure 53: Browser plugin script for Openstudy - Recommendations

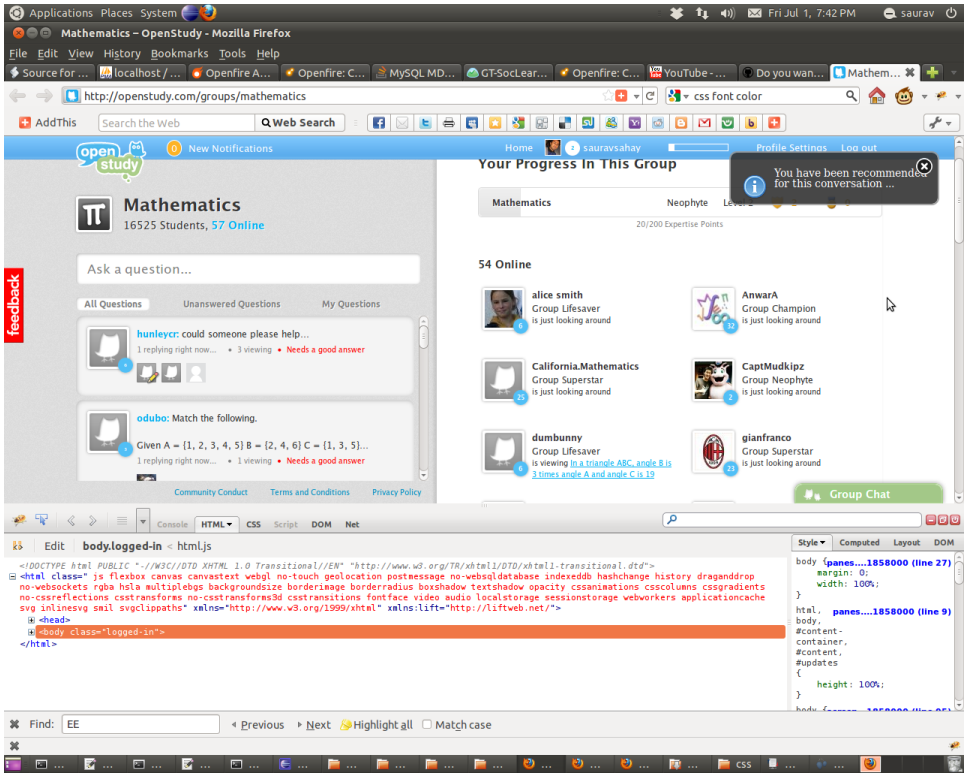

Figure 54: Browser plugin script for Openstudy - Push Notifications

CHAPTER X

WEB BASED CONVERSATIONAL RECOMMENDATIONS

10.1 Experiments

We conducted experiments to evaluate the conversational recommender for web based recommendations. We evaluated the tag guided system using Math/CS domain conversation dataset and the biomedical vocabulary guided system using the Health domain conversation dataset. We obtained relevance judgement ratings for the conversational recommendations using Amazon Mechanical Turk platform. Amazon's Mechanical Turk is a crowd-sourcing marketplace in which anyone can post tasks to be completed by paying for it. These micro-tasks, also known as Human Intelligence Tasks, are chosen by 'workers' to be completed. Once the worker has completed the task, the requester accepts or rejects the task and can also initiate further dialogue with the worker if needed. There are several ways to interact with this infrastructure. The success of this method of experimentation lies in how clearly one has created these micro-tasks, how well they have explained what needs to be done and provided enough information for the worker to complete the task. Also, the requester can require certain types of qualifications on these tasks. For example, we only wanted workers from U.S. region to complete our tasks since we thought this group could easily understand the language in the task and the problem context.

10.1.1 Datasets

For evaluation of web recommendations, we used real user generated data from an education learning community site called Openstudy. The site enables real time

conversational interactions between students/members to post questions, receive responses and study together with other students. We used our web based widget to collect data from the site by fetching conversations and populating cobot database after processing them. We collected data from several study groups on the site with different activity levels. We divided the dataset into three classes corresponding to general keyword based recommendation system, tag assisted recommendation system (tags from StackOverflow data) and ontology assisted recommendation system (using UMLS based biomedical ontologies). We extracted conversations from the following study groups:

- Tags: *Mathematics*
- Ontology: *Biology, Chemistry and Health Sciences*
- Keywords: *Art and Design, Communications and Media, Writing, Language and Culture*

Table 5: Dataset

Type	#c	#words	#episodes
Ontology	77	18.2	2.12
Social Tags	119	16.8	3.37
Keywords	62	21.44	1

where #c is the number of conversations, #words is the average length of words in conversation and #episodes is the number of interactive recommendations generated by cobot on average for each conversation.

Table 5 suggests that the general category questions from Arts domain for which cobot created simple keywords based queries from conversations were slightly longer than others on average followed by the Ontology based recommendation data which

varied from Biology to Chemistry to Health Sciences domains. We sequentially retrieved conversations from the site on a particular day without any bias to the kind of conversations being retrieved from the site. For the keyword based data that we used as our baseline, we switched off the interactive mode recommendations thus the length of these conversations is noted in Table 5 as 1.

10.1.2 Experimental Setup

We processed the data in the form shown in Figure 55 and sent them to Amazon Mechanical Turk (AMT) for annotation. We created 3 assignments for each conversation that contained few recommendations at different rank positions and at different conversation depths. The question in the conversation generated a maximum of 3 recommendations and responses in conversation generated a maximum of 1 recommendation. We limited the recommendations to this small number as we didn't want to inundate and distract the user in her conversation and focus all her attention to the web recommendation results. As shown in Figure 55, we asked the workers to rate the recommendations on a scale of 1 to 5, with 5 being 'Very good' or 'Informative and Helpful' recommendation and 1 being 'Very bad' or 'Where did this come from?' relevance judgement.

Our goal in these AMT experiments was to assess the quality of the recommended documents in conversations. We plotted graphs for the average ratings provided by the AMT workers, the best recommendation rating out of the three recommendations we suggested to the users and interactive recommendations at different trigger points in the conversation. We also modeled the task from Information Retrieval perspective and calculated the Mean Average Precision (MAP) scores for the data we had processed. For computation of MAP values, we assumed all recommendations to be relevant if they got an average rating value equal to or above 3 in the conversation.

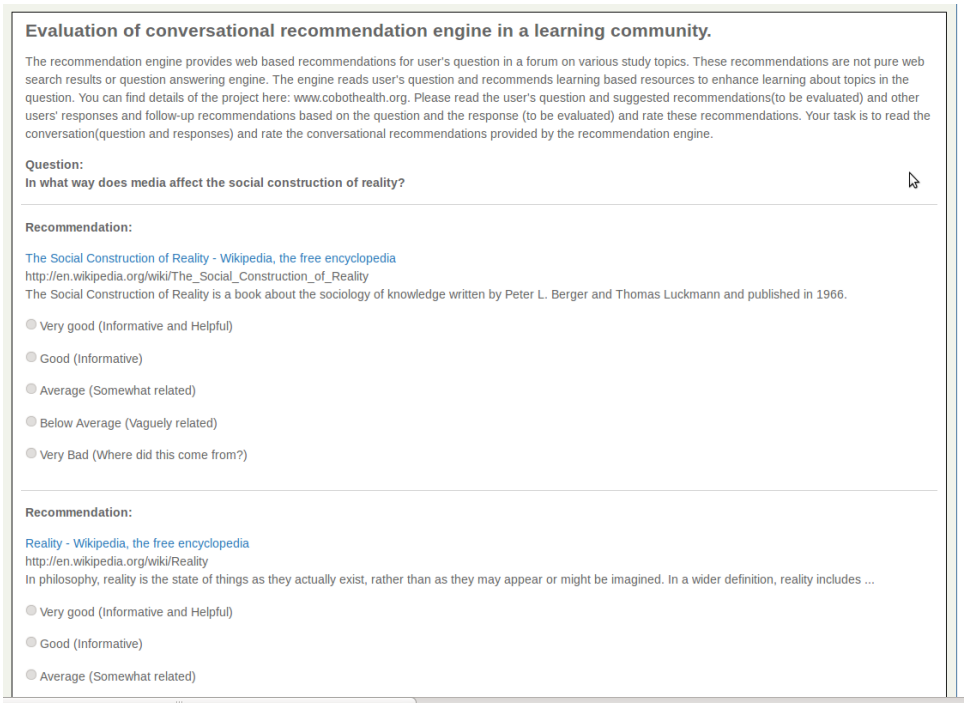

Figure 55: Human Intelligence Task (HIT) for worker

10.2 Keyword based Recommendations

We created a generic recommendation mode in cobot where we switched off the concept recognition system and relied on extracted keywords (after removal of stopwords) to trigger the cobot pipeline and generate recommendations. Switching off the concept recognition system (tags or biomedical concepts and their semantic types) also did not trigger the relation extraction and query generation pipeline since these modules bootstrapped on the recognized concepts for determining focus, relations and linguistic pattern based queries. We tested the quality of the recommendations using this mode on the Keywords dataset. Figures 56 and 57 show the best and mean ratings for different horizontal conversation lengths and overall ratings.

We also calculated the Information Retrieval based evaluation metric for the ranked retrieved conversational recommendations in the system. We calculated Precision@1, Precision@2 and Precision@3 values and the average precision values to get a final Mean Average Precision (MAP) value for the system.

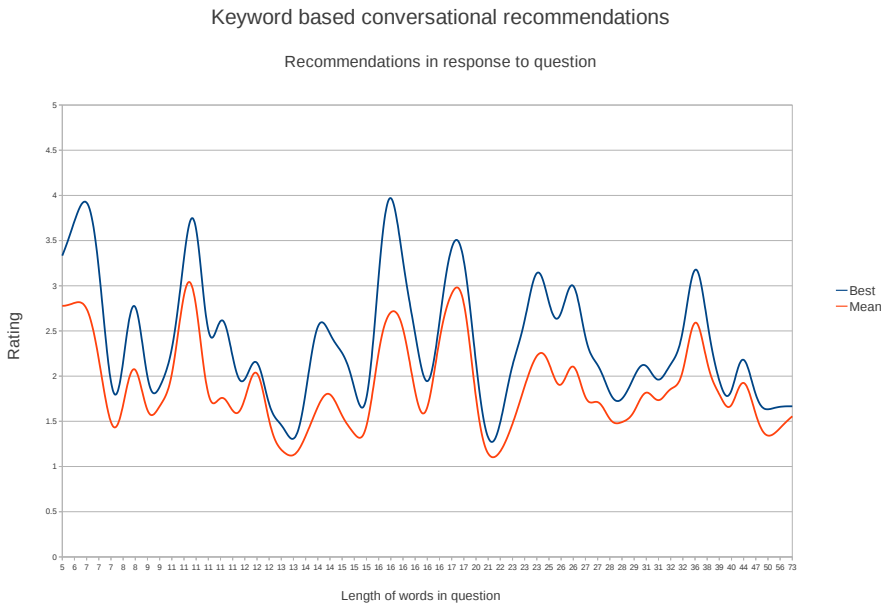

Figure 56: Keyword based recommendations - Lengthwise Ratings

We see that the results tend to be very poor in this scenario as general search engines do not retrieve very good results for long natural language sentences and cobot retrieves and processes documents from bad search indexes which result in many irrelevant ('where did this come from?') category recommendations.

10.3 Ontology guided Web Recommendations

The ontology based recommendation modules are the most complex modules in cobot with millions of terms in it's vocabulary along with semantic types and synonyms of the biomedical terms in it's knowledgebase. For this system, we fetched conversations from three different biomedical domains including Biology, Chemistry and Health Sciences. These conversations were the most subjective and domain specialized conversations in the system being asked by students taking online courses for these subjects in graduate study programs.

Figures 59, 60, 61, 62 and 63 depict the best and average ratings for biomedical conversational recommendations at different conversation horizontal and vertical

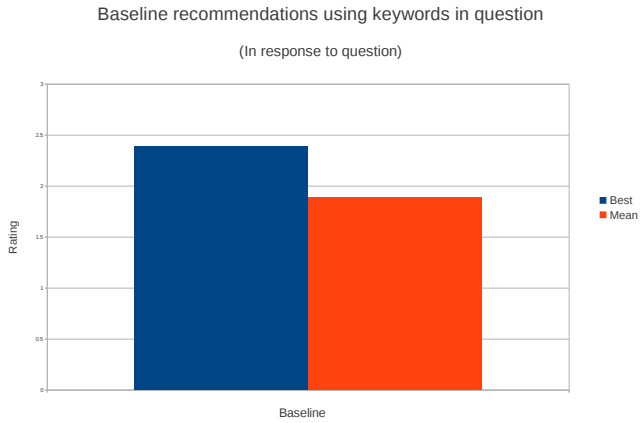

Figure 57: Keyword based recommendations - Generic Mode

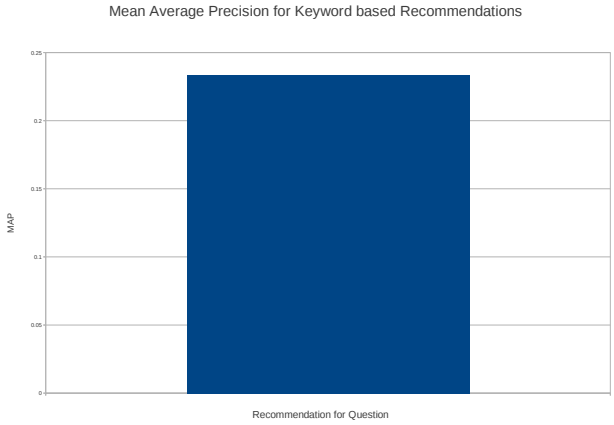

Figure 58: Generic Mode MAP

lengths.

10.3.1 Information Retrieval Metric

Figure 64 shows the MAP value for the ontology guided recommendation system.

10.3.2 Ablation Study

The goal of this experiment was to understand the individual contribution of the ontological support in cobot recommendation engine. We repeated the same experiments for the health/medical conversations, this time not using the ontology but the extracted keywords from the conversations for query formulations and search. For

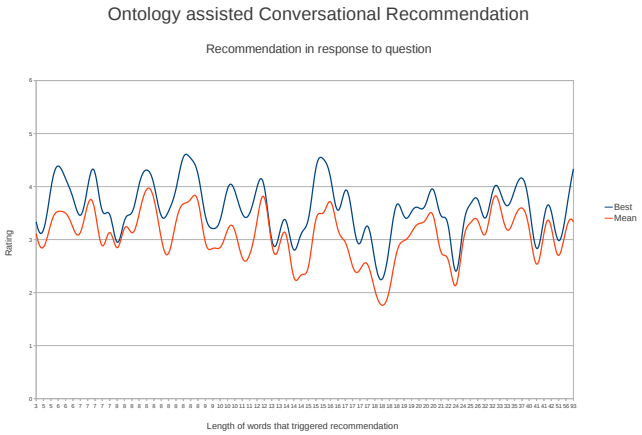

Figure 59: Ontology based recommendations for question

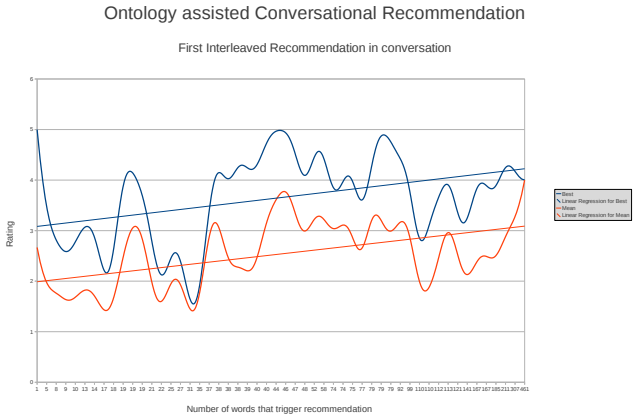

Figure 60: First interleaved Ontology based recommendations

conduction this experiment, we generated the recommendations on the same dataset that we used for ontology assisted recommendations and got similar ratings for AMT workers for the ablation study. Note however that rest of the modules remained intact in cobot (like recommendation filters such as speech act filters etc). Figure 65 and 66 show the overall average ratings and MAP score based results of the Ablation Study on the tags dataset.

It is interesting to note that ontology support didn't enhance the performance in cobot for short conversations but contributed to increased scores for medium and long conversations.

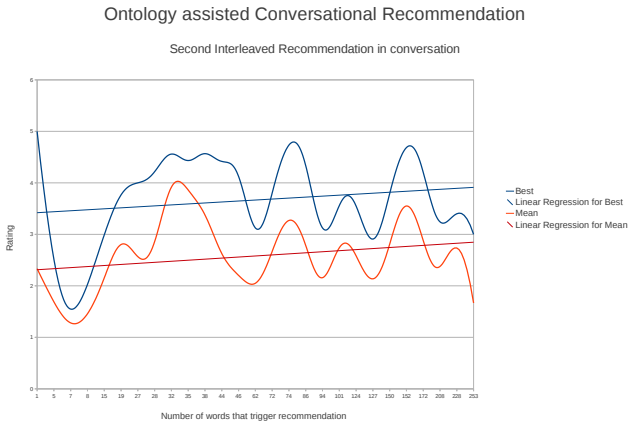

Figure 61: Second interleaved Ontology based recommendations

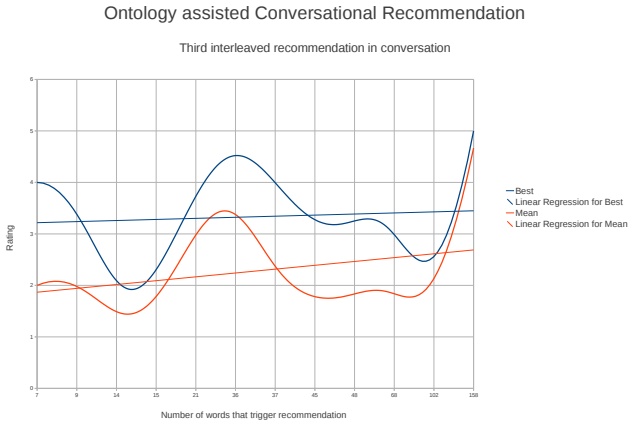

Figure 62: Third interleaved Ontology based recommendations

10.4 Tag assisted Web Recommendations

Figures 67, 68, 69, 70, 71, 72 depict similar relevance judgement scores by AMT workers on conversations from Mathematics domains and different lengths and different interactive levels.

10.4.1 Information Retrieval Metric

Figure 8 shows the MAP value for the ontology guided recommendation system.

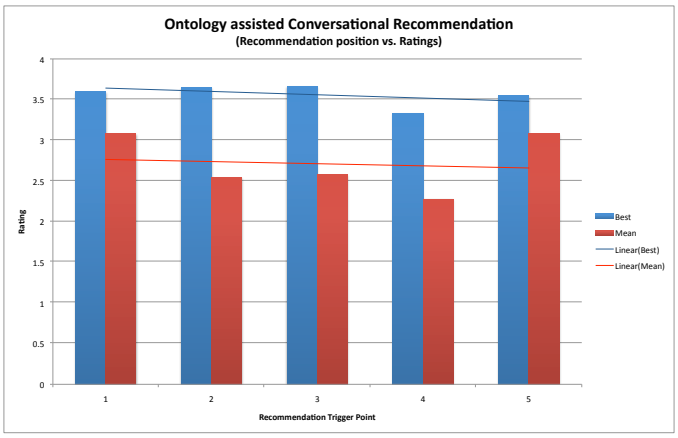

Figure 63: Ontology based recommendations - Overall

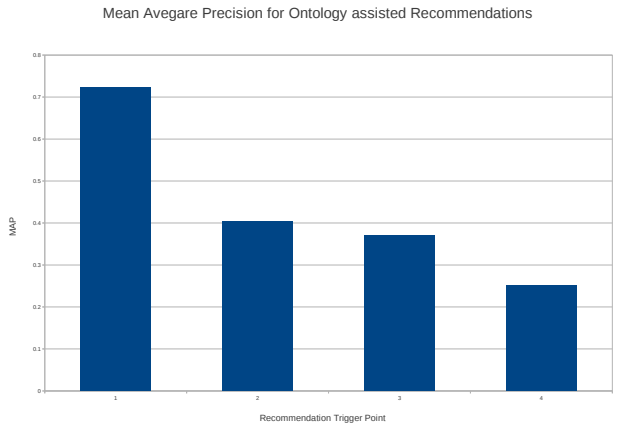

Figure 64: Ontology based recommendations - MAP

10.4.2 Ablation Study

We repeated the same experiments for the Math/CS conversations, this time not using the tags but the extracted keywords from the conversations for query formulations and search. As with the case of Medical/Health conversations, the rest of the modules remained intact in cobot (like recommendation filters such as speech act filters etc). Figure 75 and 76 show the overall average ratings and MAP score based results of the Ablation Study on the tags dataset.

It is interesting to note that tags support enhanced the performance in cobot for both short and medium sized conversations.

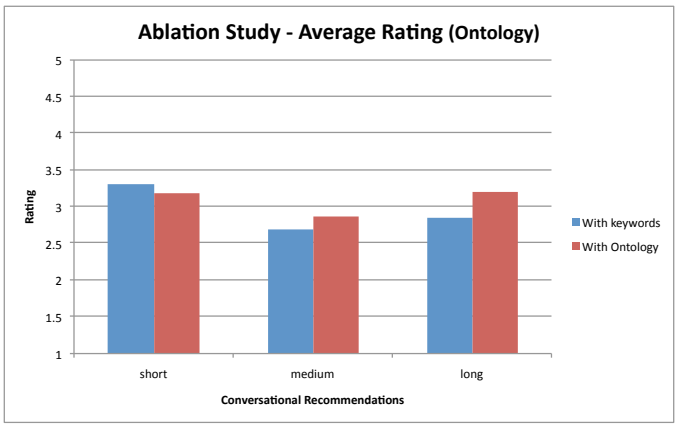

Figure 65: Ablation Study - Ratings

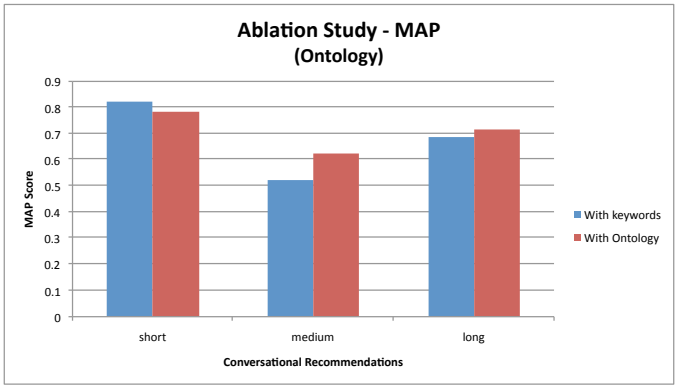

Figure 66: Ablation Study - MAP Scores

10.5 Summary

We grouped the conversations into three sets - 'short' with less that 15 words, 'medium' with less than 30 words and 'large' with greater than or equal to 30 words. Here we summarize the above results for the different conversational recommendations grouped by conversation lengths.

Table 6: Keywords based recommendations - Summary

	Avg. Rating	Best Rating	Average Precision
short	1.79	2.288	0.30
medium	1.88	2.45	0.21
long	1.82	2.10	0.14

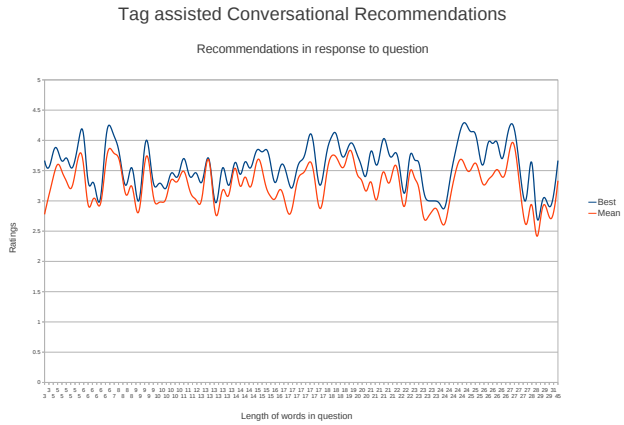

Figure 67: Tag based recommendations for question

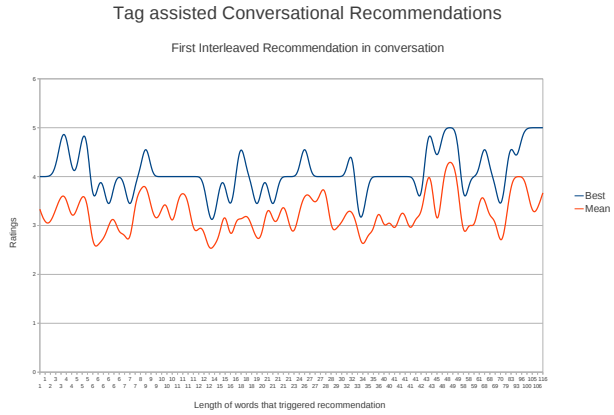

Figure 68: First interleaved Tag based recommendations

Figure 77 and 78 summarize the results of the web recommendations for our three datasets using keyword extraction method, ontology assisted method and tags assisted method as the knowledge extraction mechanism in the three studies.

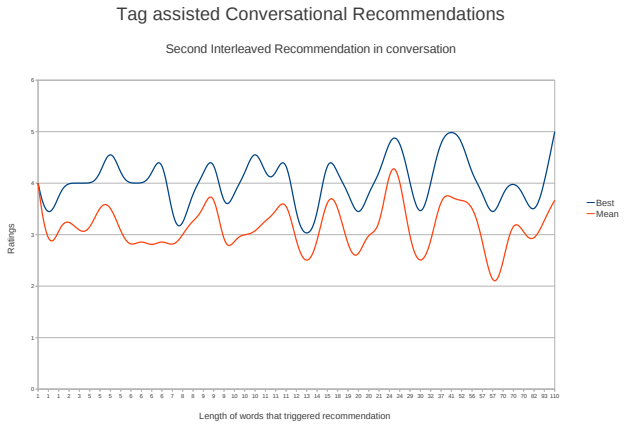

Figure 69: Second interleaved Tag based recommendations

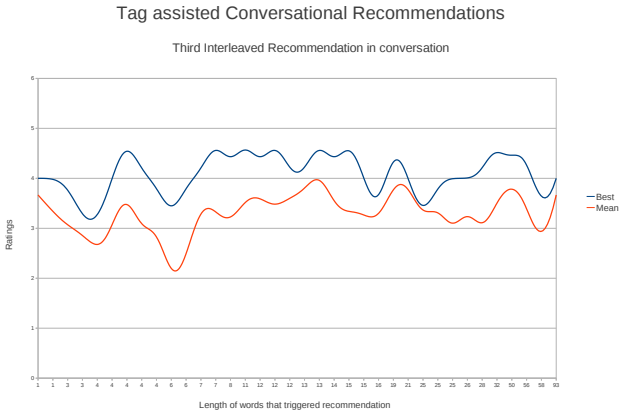

Figure 70: Third interleaved Tag based recommendations

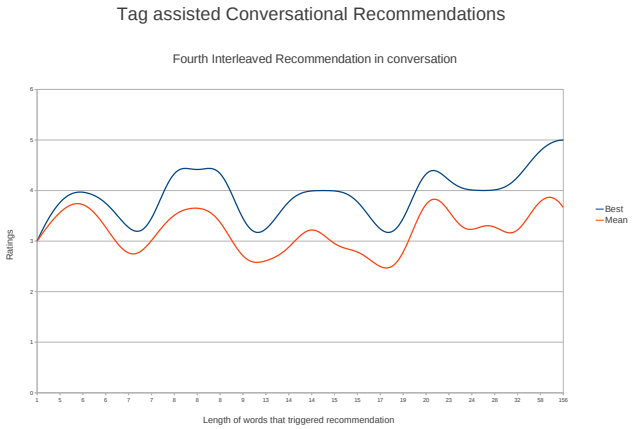

Figure 71: Fourth interleaved Tag based recommendations

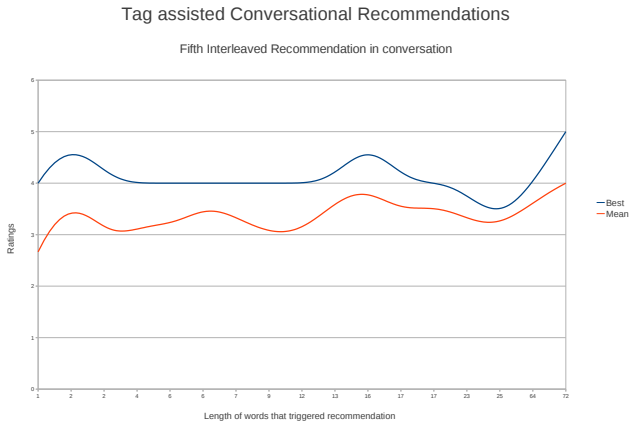

Figure 72: Fifth interleaved Tag based recommendations

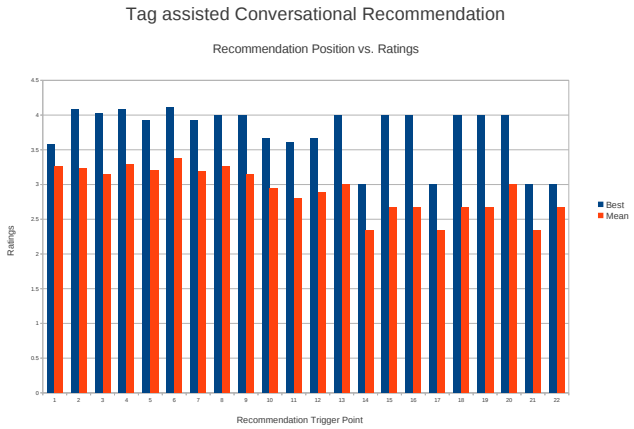

Figure 73: Tag based recommendations - Overall

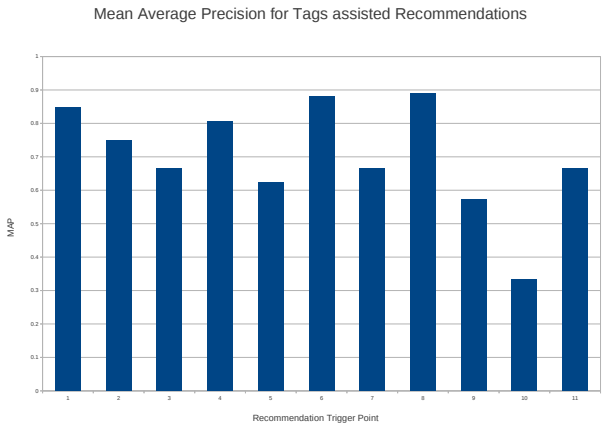

Figure 74: Tags based MAP

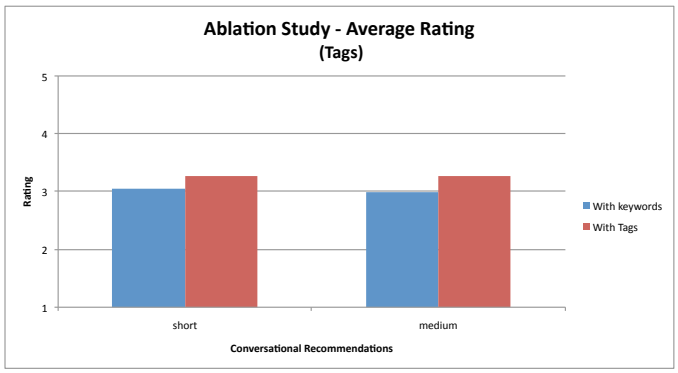

Figure 75: Ablation Study - Ratings (Tags)

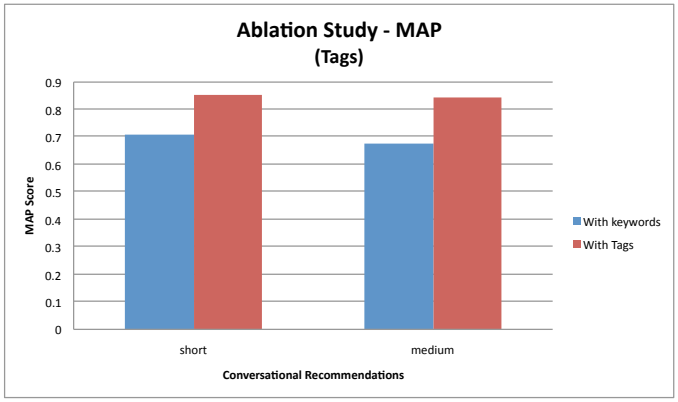

Figure 76: Ablation Study - MAP Scores (Tags)

Table 7: Ontology based recommendations - Summary

	Avg. Rating	Best Rating	Average Precision
short	3.18	3.68	0.78
medium	2.86	3.45	0.625
long	3.20	3.58	0.71

Table 8: Tags based recommendations - Summary

	Avg. Rating	Best Rating	Average Precision
short	3.27	3.53	0.85
medium	3.26	3.62	0.84
long	-	-	-

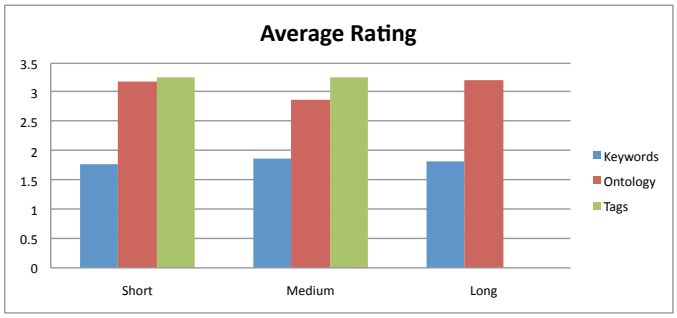

Figure 77: Average Ratings - Web Recommendations (Overall)

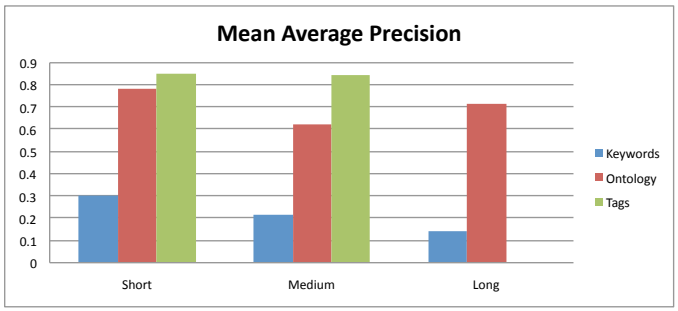

Figure 78: MAP - Web Recommendations (Overall)

CHAPTER XI

USER MODEL AND SOCIAL RECOMMENDATION EVALUATION

In User Model Adaptation, information about the learner (user) is evaluated and updated, if needed, with every system episodic interaction. This process requires a continuos addition and/or removal of user model knowledge, knowledge about the concepts, number of times they occurred, when they occurred, concept co-occurrences, associations developed, unlearning and decay with time, etc. Since learner characteristics are not constant properties, a change over time has to be considered by the learner model. We described our cognitive user model learning architecture in the User Modeling section earlier. In this chapter, we take a closer look at the user models with respect to it's ability to recommend users for conversations in the system. Since our social recommendation infrastructure is closely tied to the user models, an effective social recommender would imply that the user models have the ability to capture and update the models well to make good social recommendations in the system.

11.0.1 User size vs. Recommendations

We wanted to understand the cumulative growth of the number of recommendations with the cumulative growth of unique users in the system. We wanted to assess whether the system was able to keep up with learning new profiles as users were being added to make continuous recommendations with time. We ran a 'Playback Experiment' on the Openstudy conversation data and plotted the growth of total number of users in the system with time vs. total number of recommended users

with time.

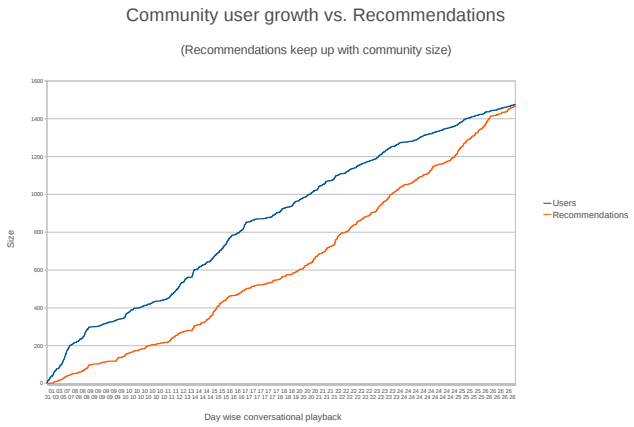

Figure 79: Community size vs. User Recommendations

Figure 79 depicts our temporal playback experiment plot and shows that the system was continuing to make user recommendations with different snapshots and thus giving us a hint that it was recommending users for new conversations and new users based on the profiles it learn from previous user interaction episodes.

11.0.2 Explanation

In this section, we will explain how the user model filters contribute to the overall social recommendation pipeline along with contribution of other filter components. As our first step, the search index that had indexed all users on the extracted concepts, relationships between concepts and the associated conversation text returned some candidate users matching on the generated queries coming from conversations. After this initial candidate generation, other filters, along with the user model filter was applied to the candidates to pick up final list of users to recommend. In Figure 80, we show a list of 9 users that were recommended for a particular conversation and the contribution of different scorers in the system. We see that the top users returned by our search index (with highest Lucene Hit scores) did not contain match with any of the concepts in their short term or long term models. Instead, some users down the

list had matches in their short term models as well as spreading activation matches in their long term models suggesting that these users had recently spoken about the concepts in conversation and they were also generally interested in the concepts from these conversations.

	Rank	Score	STM	LTM	Model	Bonding
Breanna025	1	0.3369116	0.0	0.0	0.0	0.0
ks1007	2	0.28479117	0.0	0.0	0.0	0.0
emf.22.soccer	3	0.22662348	0.0	0.0	0.0	0.0
tedwilcox	4	0.21708477	0.0	1.8	0.72	0.0
flamist91	5	0.21708477	0.0	0.0	0.0	0.0
roldy	6	0.20405468	0.0	0.0	0.0	0.0
Woodrow	7	0.18602645	0.0	0.0	0.0	0.0
toni32	8	0.14414129	0.0	0.0	0.0	0.0
MD	9	0.14239559	1.0	1.8	1.32	0.0

Figure 80: Scoring for User Recommendations

11.0.3 LTM visualization

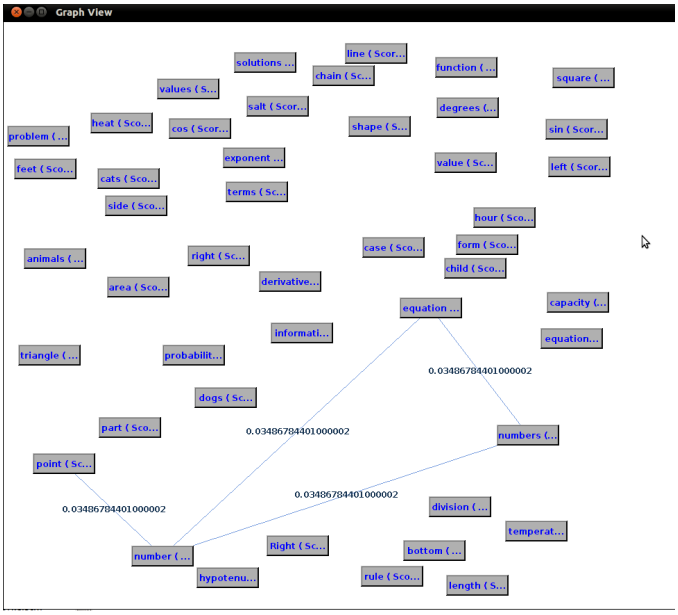

Figure 81: Long Term Model Visualization

We wanted to examine how the long term user models were learning new user

concepts and developing associations between them with time. Figure 81 shows one user that we snapshot at a particular instant in the playback. The graph shows the different concepts along with the strengths on these concepts and the associations developed between concepts as a result of their frequent co-occurrentces in different conversations that this user was associated with.

11.0.4 Social Capital Contribution

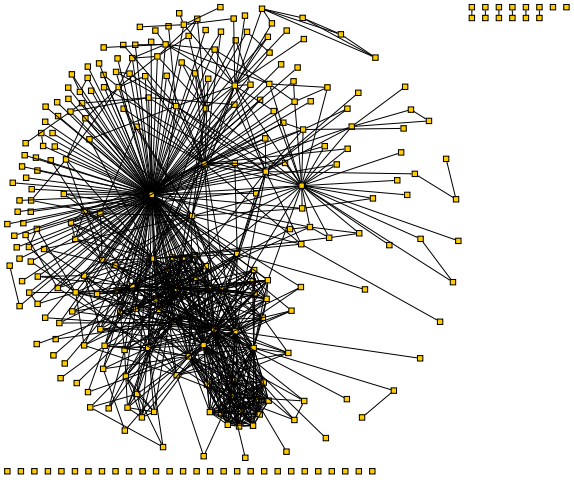

Figure 82: Community Implicit Capital

We plotted the social bonding interaction network on our dataset as shown in Figure 82. This graph shows the user interaction network in the system. We observe that there were few users in the system that had strong bonding networks while others had had few interaction episodes with shared concepts together that created bonding capital between them. Therefore, this module contributed scores towards picking up users who had spoken about similar concepts with each other before. We think of this module as being very important in real community deployed systems since it is a common phenomenon that users develop social bonds with others through interactions and may continue to do so because of the social bonding.

CHAPTER XII

USER STUDIES

12.1 Preliminary Experiments

We recruited four subjects during our initial design stages of cobot system. We developed a system prototype with focus on interaction workflow and design (Figure 83). These subjects were graduate students of Georgia Tech and our purpose was to get an initial feedback to understand if we were going in the right direction. We called our web based system prototype Healthbuzz. We conducted surveys, interviews and a usability study of the Healthbuzz system. We created two different scenario-based tasks and asked the subjects to perform the tasks on Healthbuzz and Web Search engine. Since students were concerned about H1N1 and related questions, we created our tasks accordingly.

Figure 83: Cobot Interface

Scenario 1: *You are a GT student living in the dorms. You believe your roommate*

is showing some H1N1 symptoms including: coughing, having a runny nose, and fever. How can your roommate find out if he has been infected with H1N1? What services does GT offer regarding H1N1 prevention? What can you do to prevent yourself from getting infected?

Scenario 2: *You are a GT student living in the dorms. You believe that you have become infected with H1N1. What are the treatments offered by GT? What does GT advise you to do (including class schedules) if you become infected?*

Besides interviews, we asked users to rate on 3 different parameters. The averaged results are given in Figure 84 for the 3 questions where 3 indicates Strongly Satisfied.

- The level of satisfaction with time spending on finding the results you want. (Timeliness)
- The level of satisfaction with the results you find. (Results)
- How easy it is to use the system? (Ease of Use)

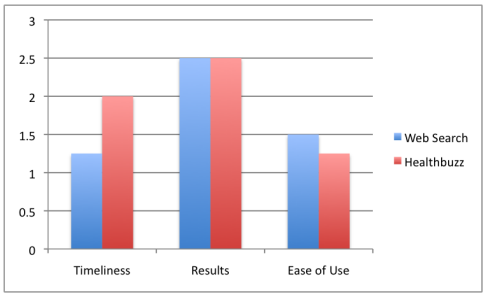

Figure 84: Evaluation Results

First, we conducted a 'within subject' experiment for our user testing. Our participants were asked to perform scenario-based tasks using Healthbuzz and web search. The usability of the Healthbuzz was evaluated through quantitative methods by using a 7-point Likert scale survey. This survey was used to ask the participants' level of satisfaction with the time they spent, the results they got, and the easiness of performing the tasks. The primary intent of this usability testing was to determine if our

participants could perform the assigned tasks and their subjective level of satisfaction with using the systems.

12.1.1 Interviews

During the initial analysis of the study results, it was noticed that there were recurring themes presented across study participants. Upon noticing this, the major themes were extrapolated and the study data was then analyzed according to those major themes. The results of those analyses are presented in the following sections.

12.1.1.1 Interaction Method

Participants of the study enjoyed the use of chat conversations to gather information. It was noted that some users valued the ability to create a group chat by inviting others to the conversation. The following participant quotes echo the feelings expressed throughout the study: "I feel good about having a conversation with someone." "I liked the conversation part." "I liked talking with people. I liked the group conversation with more than one person." Participants stated that the system seemed to center around building social networks of people to talk with. One participant stated that, "The reason to come back is being able to connect with other people."

As well as interacting through chat with participants, HealthBuzz also gave web recommendations based on contents of the chat conversation. Participants liked the information provided by the web recommendations and felt that it added legitimate health information to the conversation. However, participants also indicated that at certain times there were too many recommendations and the results became confusing. Also, some noted that recommendations given by HealthBuzz should have been more refined and discriminating. A participant said, "I liked the automatically updated web recommendations. I just want the results to be more refined. I typed 'thank you,' and it gave me thank you related responses." Participants also indicated that they liked the single destination point to find the health information that they needed.

This is as opposed to the web search which required users to navigate many different web pages. A participant noted that, "I like the combination of chatting, forum (searching old conversations), and web search functions (web recommendations) all together. Otherwise I would have to go to three different places."

12.1.1.2 Privacy

Due to the nature of the information being exchanged within the HealthBuzz system, privacy was a major concern expressed by all participants. Participants and perspective users of the system stressed the need for anonymous communication when discussing health related issues. A participant stated, "I want to see a privacy protected function. Anonymity should be promised." Participants wanted the ability to switch between being anonymous and using their screen name.

As well as contributing to a conversation anonymously, participants stressed the need to be able to mark entire conversations as private, allowing only approved users to contribute or search for the conversation. "I want to see a private conversation function. At the beginning of the conversation, I want to choose this conversation to be private or public." Individuals especially indicated the need for anonymous communication when discussing health issues with associated negative connotations, such as sexually transmitted diseases.

12.1.1.3 Authority and Accuracy

A main topic brought up by some of the participants, was the accuracy of the information being given in conversations. Participants seemed somewhat satisfied with the results given by the web recommendations, as the majority of the recommendations came from reputable health information sites. The main concern centered on the other users of the system and the lack of accountability and authority. A participant indicated this in their statement, "The recommended people list only provides their name. I want to see more about their info before I decide to invite or talk to the

person." Without more information about users of the system, participants expressed concern as to the validity of the information that would be conveyed.

One participant summed this up these feeling of authority and accuracy in their interview. "I have a strong opinion about asking medical related info on Web. My dad and brother are doctors, so whenever I have health related question I call them. I would never ask my personal health related questions on Web. Even if I found some information, I would not trust it unless it's from a legitimate source like WHO. I think health info is very sensitive info. I think anybody who doesn't have a lot of medical knowledge can't recommend medicines. I don't think I would trust any info on the Web posted by random people."

12.1.1.4 Time

Some users of the system were concerned about the amount of time it would take to find the health information they were looking for. These concerns centered around two topics: the interaction method of chatting and the response time needed to answer questions. Concern was noted by a participant that the method of interacting through chat would take more time than doing a web search. In their words, "Chatting is for people who have time to spend."

Another concern that centered on the chat method of interacting was that another user would not be online to interact with. There was concern among the participants that there would not be users online with which they could interact with. Participants expressed that they would not wish to start a new conversation if they would have to wait a long amount of time for people to join. Although not a forum system, the following quote reflects a view that is relatable to HealthBuzz, "I have used forums for web developers. I have posted a question and it took 2 days to get the right answer." It is interesting to note that users were worried about the method of chat interaction and the amount of time spent, but still expressed that they enjoyed this

type of interaction.

12.1.2 Analysis

Based on interviews with students, we found that web searching is the most frequently used method when looking for information related to health issues. However, students addressed that it is difficult to find a place for discussing local health related issues with other students. They are familiar to finding general information by using a search engine (e.g., Google) or a health portal (e.g., WebMD). However, it remained a challenge to find specific information that exactly fitted their particular issues (eg. How many cases of H1N1 have been reported in the Northside dorms of graduate housing?'). For specific health related issues they wanted to discuss, they found that Web forums are useful since they can post their specific questions and interact with other people. We noted two major shortcomings with Web forums. Since the forum is usually available for anyone, there are trust issues. Another problem comes from its asynchronous characteristic. Users have to wait until other people post their responses to their questions. When they have a follow up question or want to interact with the person, they need to post further questions or comments and wait for the responses again. It is difficult to expect even near real time interactions due to the asynchronous nature of the medium. Generic IM chat provides very limited friend list based on one's personal social network. Also, it is difficult to find a person who would be knowledgeable about their health related issues among their IM friends list. Also, when it comes to discussing specific and personal health related problems (against generic information), one is not comfortable discussing these issues with the people they know.

What we learnt from the interviews is that people connections were a crucial aspect of Healthbuzz system. Collaboratively seeking information with similar users lead to more refined and satisfied search experience than solitary Web search. We

had thought that providing Web based results in a conversation would be a crucial aspect of the system. This came out as a surprise to us when users' did not show an active interest in the web based recommendations alongside in the conversation. Unless pointed to explicitly by one of the participants, the users generally didn't pay attention to what was showing up as the recommended articles. We also realized that we needed to restrict these recommendations to just a few really good conversational recommendations when it was required.

Participants of the study enjoyed the use of chat based conversations to gather information. It was noted that some users valued the ability to create a group chat by inviting others to the conversation. Due to the nature of the information being exchanged within the Healthbuzz system, privacy was a major concern expressed by all participants.

We learnt that the socially powered search and the ability to collaboratively search together and solve issues with real people(instead of solitary queries on search engines) is a very powerful medium and is highly contextual. Websites like Vark.com are doing this effectively using Instant Messaging(IM) based messaging bots. People spent more time on Healthbuzz system as compared to solitary web searches, exploring more questions and arguments, engaging naturally into the conversation accepting and sending conversation invites by/to others. We also learnt that a Healthbuzz like system has many other applications revolving around sharing experiences, engaging the local community into a dialogue about different issues, events and opinions.

12.2 User Studies on Widget

Figure 85 shows the results from 10 qualitative user studies with the Openstudy widget prototype. The most striking feature that people liked about this recommendation engine was the Diversity feature and the ability to see exploratory recommendations relevant to the conversation while in the conversation. Most other responses were

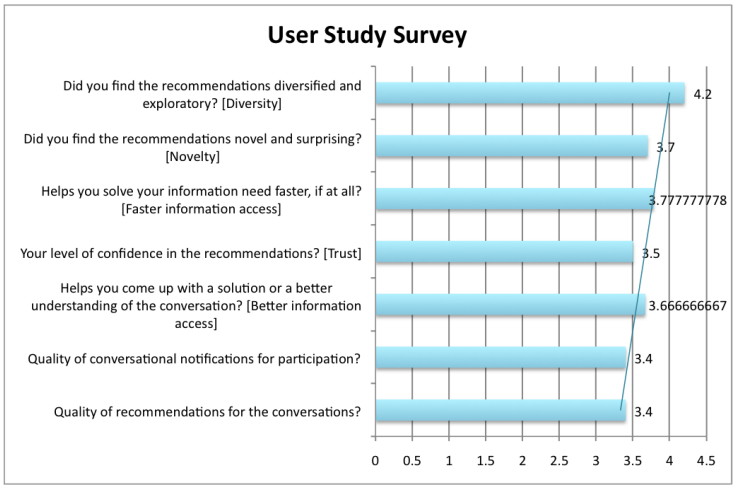

Figure 85: Survey Results - Recommendations

either 'good' or 'adequate' in the system.

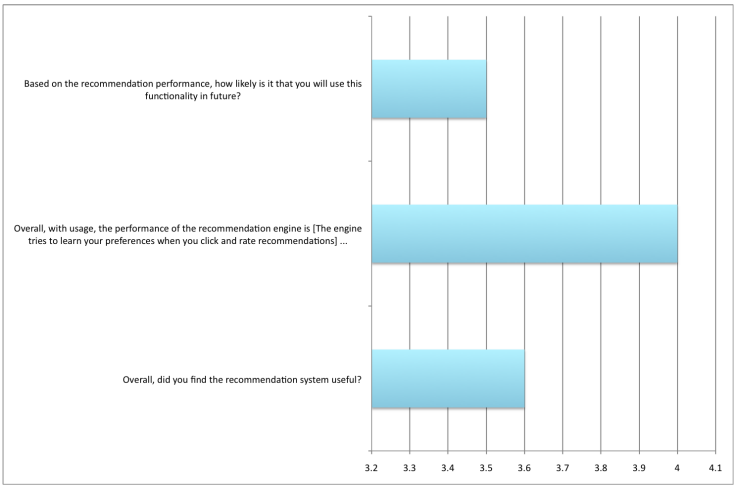

Figure 86: Survey Results - Overall

One interesting result we found, as shown in Figure 86, was that everyone reported that the system was getting better. This was the second best response, the first being that the system was getting much better with time. This suggested that the users seemed to like the incremental updates in the recommendations as well recommendations coming out of a larger related pool of community recommendations.

We also received some qualitative feedback from users who asked questions, rated conversations and checked out other existing conversations on the site[87]. People seemed to like the natural language aspect and the system's ability to recommend

Survey Results

The recommendation system is really useful for long and detailed questions that people might have. For a straight forward question, e.g. what is diabetes or what is cancer, a regular internet search engine may appear to be more effective. However, if one has a long, detailed and a specific question, like 'I am diabetic and have recently felt severe loss of appetite after taking my medicines, is there a need to go to the doctor', that is where this engine is able to direct the person to something very relevant.

works great with specific questions (which appeared to be identifiable keywords) but was not able to come up with relevant suggestions in case where the questions lacked keywords or were slightly vague. However with repeated usage looks like the system is able to correctly find the pattern/interests/required domain of information and will be able to suggest very accurate/relevant information as per my requirement.

I like the way the system recommends answer to my questions. The more questions I ask related to any questions the more related answers I get from the system. I feel that if the system is tweaked in the right way then it could become a very powerful recommendation engine for users. Looks like it is a work in progress and I would love to see a much more powerful algorithm to be implemented on this system for generating much more relevant answers to users questions.

The recommendations seems good to me. But, there are scope of improvements. For, example when I posted the question 'What is the function of haemoglobin' . The Cobot system gave some relevant recommendations but I also suggested content related to Mathematics which may be because the query contained word 'function'. Overall, the recommendations is great.

a) Cobot helping me connect to other people is very helpful. I can find the experts in the area in which I have typed the questions. b) Typing in natural language is very convenient. c) I dont have to go and look for the websites where I need to go and look for information d) If the information is embedded in various websites, i dont have to go in and search for the information on the websits, Cobot helps me find that embedded information

The unique feature about this system is that it allowed me to pose my queries as a descriptive sentence rather than some specific and relevant keywords. This is helpful because sometimes I need to search for a concept for which I do not really know the right keywords that will fetch me the relevant results. The system was intelligent enough to map my queries (presented in plain English sentences) into the right context of my real intention and generated adequately relevant recommendations. Sometimes it also generated the exact technical representation of my very general query (such as mapping a set of symptoms to a set of probable diseases causing those symptoms). However, the quality of the recommendations seem to vary across the various domains. In particular, I found better recommendations for biology or health related queries. Even then, I feel that the system can be much improved in these areas too.

I think more data in the system would help to find more relevant content for people's questions.

I felt the conversation recommendations were actually pretty good. Web resource recommendations were not bad.

More external forum links than common web recommendations such as yahoo and wikipedia

it is novel and very exciting to see an automated engine work in a humanly helpful way.

Figure 87: Survey Results - Text Responses

relevant pages in large conversations.

CHAPTER XIII

SUMMARY OF CONTRIBUTIONS

We present a summary of the main contributions accomplished in this dissertation. We proposed a novel socio-semantic community based conversational recommendation system and ideated that such a system would help solve user's information access problem better and with less effort. We conducted several experiments, evaluations and user studies with the system and got a mixed bag of not so good, good and encouraging results.

We organize our contributions and impact along the following three primary dimensions and sources of power for the thesis, i.e. a blended recommendation environment, knowledge based information architecture and evidence based recommendations.

- *Blended recommendation environment.* Cobot provides a unique blended recommendation environment, one that helps in multi-modal recommendations and thus blended learning outcomes for conversational content. It also helps in reduction of additional search effort due to the system's ability to extract multiple queries automatically and bring in recommendation results. Recent advances and successes in blended learning models in Education domain also suggest that 'blended learning' is more effective than traditional learning alone, especially when the blended learning experiences are well designed. There are two main components of a successful blended learning experience. One is access to consistent and reliable online content, and the other is contextual and timely human interaction. Cobot provides both components and therefore has the potential to be a practical recommendation system providing content access

and an engaging experience through expert social interaction together.

- *Knowledge based information architecture.* Cobot is a knowledge based domain adaptable information processing system. It bootstraps on the recognized evidential knowledge to trigger downstream extraction components for generating queries for semantic search, indexing of extracted data, user model update with extracted knowledge and semantic filtering or candidate results for final recommendation generation. The main advantage of this approach over purely statistical approaches center around having a handle on precise knowledge in recommendation candidates for applying different usecases for integrated reasoning, decision support and problem solving. With proper engineering effort, a knowledge based system with lexico-syntactic rules and patterns, effecting parsing and applied reasoning can result in very effective decision support recommendation system. Our goal in this work was not to constrain the agent to a particular sub-domain like diseases, treatments or drugs related conversations, etc. but to stay as generic and automated as possible using large controlled vocabularies such as the UMLS ontology for generic domain recommendations using social conversations.

- *Evidence based Recommendations.* Cobot's ability to bring in new knowledge from external web data, process and extract that knowledge and use it as past cases while scoring and evaluating new candidates against the conversation makes it an experience gathering recommendation system. This also gives it the ability to promote community preferred results by using the community preferences as an evidence filter for final recommendation generation. Cobot retrieves candidates in real time from trusted sources thus making sure that fresh evidences are evaluated along with past episodes for generating the final recommendations. Cobot also uses social metrics based on social capital theory

to promote such social recommendations that have had interactions in the past along with people in the conversation.

13.1 Summary of findings

- Users reported that the conversational recommendation system was useful for long, detailed and specific questions.
- Users perceived that the system was getting better with more interactions in conversations.
- Most users reported that the recommendations were diversified and exploratory.
- The ontology supported bio-medical domain recommender got best ratings (Avg. Rating: 3.2) by mechanical turk workers for long conversations (¿ 30 words) with a MAP score of 0.71
- The social tags based recommender for the Mathematics domain got overall best average ratings and MAP scores for short and medium length conversations.
- Ratings for recommendations stayed about the same or improved with increase in both length (size of question/response) and height (number of interactions in conversation) of conversation.
- Playback Experiment for User Recommendation evaluation on Openstudy data suggested that Hybrid Social recommendations considered several factors such as recency of interaction, social bonding, long term interests in topics as well as query match scores to select users for recommendations.
- for short conversations, extracted keywords (slightly) outperform the ontology based setup (on average and based on MAP scores), given every other recommendation module stayed the same in cobot.

- for short conversations, tag supported recommendations outperform the extracted keywords based recommendations(on average and based on MAP scores), given every other recommendation module being the same in cobot.
- for medium and long conversations, both tags and ontology supported recommendation system did better than extracted keywords (on average and based on MAP scores).

CHAPTER XIV

FUTURE WORK

This thesis opens up many new and interesting directions ahead to take this work forward in the following areas:

- *Living Lab Prototype* We deployed cobot as a widget on Openstudy.com and got real data into cobot for recommendation generation. We also did some user studies using the widget on Openstudy to get users test, feel and evaluate the cobot prototype on Openstudy. However, we didn't get critical mass and real users (for example, continuing Openstudy users) to continuously use the system, provide suggestions and give us feedback to improve the overall utility of cobot engine. We would like to deploy cobot in a real setting and conduct living lab experiments on it to make it a robust and improvig technology.

- *Social Search* Social Search is a very active and interesting area of research today. We did some preliminary work and came up with suggestive algorithms for community balance based on user intentions and daily interactions. We would like to deploy cobot over instant messaging channels and study the social question answering phenomenon.

- *Decision Support* We developed cobot as a generic recommender in this dissertation. We would want to fine-tune cobot's knowledge-bases to focus on certain medical issues such as disease finding, diagnostics, symptoms and treatments and work towards developing decision support recommendations for patient health.

- *Cognitive User/Community Modeling* Cobot currently has a fine-grained short

term and long term user model. How do we adapt the system to generate cognitive group models? How do we add more knowledge and processes in the cognitive models. This would be a very interesting area of research we would want to extend this work towards.

- *Language Understanding* Improving the coverage and reliability of syntactic analysis, semantic parsing and extraction in conversations with near real time delivery of results to keep the users engaged requires highly sophisticated and robust taggers, parsers and classifiers. We would extend our parsers into generic trainable syntactic expression parsers using learning and classification algorithms.

- *Mixed initiative Agengy* Cobot as a platform was developed as a mixed initiative agent so that users could tweak cobot's recommendations (as in a wiki system). We would want future algorithms in cobot to learn and pick up from such improvements so that cobot could apply heuristics to modify it's own recommendations.

REFERENCES

[1] Agichtein, E., Gabrilovich, E., and Zha, H., "The social future of web search: Modeling, exploiting, and searching collaboratively generated content.," *IEEE Data Eng. Bull.*, pp. 52–61, 2009.

[2] Agichtein, E. and Gravano, L., "*Snowball*: extracting relations from large plain-text collections.," in *ACM DL*, pp. 85–94, 2000.

[3] Agichtein, E. and Gravano, L., "Snowball: Extracting relations from large plain-text collections," in *Proceedings of the Fifth ACM International Conference on Digital Libraries*, 2000.

[4] Agichtein, E. and Gravano, L., "Querying text databases for efficient information extraction.," in *ICDE* (Dayal, U., Ramamritham, K., and Vijayaraman, T. M., eds.), pp. 113–124, IEEE Computer Society, 2003.

[5] Aha, D. W., Kibler, D., and Albert, M. K., "Instance-based learning algorithms," in *Machine Learning*, pp. 37–66, 1991.

[6] Allan, J., "Incremental relevance feedback for information filtering," pp. 270–278, ACM Press, 1996.

[7] Allen, J., Chambers, N., Ferguson, G., Galescu, L., Jung, H., and Taysom, W., "Plow: A collaborative task learning agent," in *In Proc. Conference on Artificial Intelligence (AAAI*, pp. 22–26, Springer-Verlag, 2007.

[8] And, D. B., "Revising user profiles: The search for interesting web sites," 1996.

[9] Anderson, J. R. and Lebiere, C., "The atomic components of thought," 2000.

[10] Anthony G. Francis, J., Devaney, M., Santamaria, J. C., and Ram, A., "Scaling spreading activation for information retrieval," in *Proceedings of IC-AI 2001*, July 25 2001.

[11] Aronson, A. R., "Effective mapping of biomedical text to the umls metathesaurus: The metamap program," 2001.

[12] Baffes, P. and Mooney, R., "Refinement-based student modeling and automated bug library construction," *Journal of Artificial Intelligence in Education*, vol. 7, pp. 75–116, 1996.

[13] Balabanovic, M. and Shoham, Y., "Fab: Content-based, collaborative recommendation," *Communications of the ACM*, vol. 40, pp. 66–72, 1997.

[14] Berkman, L. and Kawachi, I., *Social epidemiology*. Oxford University Press, 2000.

[15] Berland, M. and Charniak, E., "Finding parts in very large corpora," in *ACL*, 1999.

[16] Bollacker, K., Lawrence, S., and Giles, L. C., "CiteSeer: An Autonomous Web Agent for Automatic Retrieval and Identification of Interesting Publications," in *Proceedings of the Second International Conference on Autonomous Agents* (Sycara, K. P. and Wooldridge, M., eds.), (New York), pp. 116–123, ACM Press, 1998.

[17] Brin, S., "Extracting patterns and relations from the World Wide Web," *Lecture Notes in Computer Science*, vol. 1590, pp. 172–??, 1999.

[18] Brin, S. and Page, L., "The anatomy of a large-scale hypertextual web search engine," *Comput. Netw. ISDN Syst.*, vol. 30, no. 1-7, pp. 107–117, 1998.

[19] Brüninghaus, S. and Ashley, K. D., "Reasoning with textual cases," in *ICCBR*, pp. 137–151, 2005.

[20] Budzik, J., Sood, S., Hammond, K. J., and Birnbaum, L., "Context transformations for just-in-time retrieval: Adapting the watson system to user needs," 2006.

[21] Burke, M., Marlow, C., and Lento, T., "Social network activity and social well-being," in *Proceedings of the 28th international conference on Human factors in computing systems*, CHI '10, (New York, NY, USA), pp. 1909–1912, ACM, 2010.

[22] Burke, R. D., Hammond, K. J., Kulyukin, V. A., Lytinen, S. L., Tomuro, N., and Schoenberg, S., "Question answering from frequently asked question files: Experiences with the FAQ finder system," Tech. Rep. TR-97-05, 1997.

[23] Burton, R. R. and Brown, J. S., "A tutoring and student modelling paradigm for gaming environments," *SIGCUE Outlook*, vol. 10, pp. 236–246, February 1976.

[24] Callan, J., "Learning while filtering documents," in *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 224–231, 1998.

[25] Carbonell, J. R., "AI in CAI: An Artificial-Intelligence Approach to Computer-Assisted Instruction," *Man-Machine Systems, IEEE Transactions on*, vol. 11, pp. 190–202, Dec. 1970.

[26] Carmel, D., Yom-Tov, E., Darlow, A., and Pelleg, D., "What makes a query difficult?," in *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '06, (New York, NY, USA), pp. 390–397, ACM, 2006.

[27] Carr, B. and Goldstein, I. P., "Overlays - A theory of modeling for computer-aided instruction," tech. rep., AI Lab Memo 406 MIT, 1977.

[28] Chen, L. and Sycara, K., "Webmate: A personal agent for browsing and searching," in *In Proceedings of the Second International Conference on Autonomous Agents*, pp. 132–139, ACM Press, 1998.

[29] Chiu, B. C. and Webb, G. I., "Using decision trees for agent modeling: improving prediction performance," *USER MODELING AND USER-ADAPTED INTERACTION*, vol. 8, pp. 131–152, 1998.

[30] Chomsky, N., "Formal properties of grammars," *Handbook of mathematical psychology*, vol. 2, pp. 323–418, 1963.

[31] Chomsky, N. and Schutzenberger, M., "The algebraic theory of context-free languages," *Computer programming and formal systems*, pp. 118–161, 1963.

[32] Church, K. W. and Hanks, P., "Word association norms, mutual information, and lexicography," *Comput. Linguist.*, vol. 16, pp. 22–29, March 1990.

[33] Cimiano, P., Handschuh, S., and Staab, S., "Towards the self-annotating web," in *WWW* (Feldman, S. I., Uretsky, M., Najork, M., and Wills, C. E., eds.), pp. 462–471, ACM, 2004.

[34] Cimiano, P., Ladwig, G., and Staab, S., "Gimme' the context: context-driven automatic semantic annotation with C-PANKOW," in *WWW* (Ellis, A. and Hagino, T., eds.), pp. 332–341, ACM, 2005.

[35] Collier, N., Nobata, C., and ichi Tsujii, J., "Extracting the names of genes and gene products with a hidden markov model," in *COLING*, pp. 201–207, Morgan Kaufmann, 2000.

[36] Collins, A. and Quillian, M., "Retrieval time from semantic memory," *Cognitive Psychology: Key Readings*, vol. 2, p. 395, 2004.

[37] Converse, T., Kaplan, R. M., Pell, B., Prevost, S., Thione, L., and Walters, C., "Powerset's natural language Wikipedia search engine," in *Wikipedia and Artificial Intelligence: An Evolving Synergy, Papers from the 2008 AAAI Workshop*, 2008.

[38] Cronen-townsend, S., Zhou, Y., and Croft, W. B., "Predicting query performance," 2002.

[39] Cross, R., Rice, R. E., and Parker, A., "Information seeking in social context: structural influences and receipt of information benefits," *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, vol. 31, no. 4, pp. 438–448, 2001.

[40] de Marneffe, M.-C., Grenager, T., MacCartney, B., Cer, D., Ramage, D., Kiddon, C., and Manning, C. D., "Aligning semantic graphs for textual inference and machine reading," in *Proc. of the AAAI Spring Symposium at Stanford. 2007*, 2007.

[41] de Marneffe, M.-C., MacCartney, B., and Manning, C. D., "Generating Typed Dependency Parses from Phrase Structure Parses," in *Proceedings of the IEEE / ACL 2006 Workshop on Spoken Language Technology*, The Stanford Natural Language Processing Group, 2006.

[42] Dill, S., Eiron, N., Gibson, D., Gruhl, D., Guha, R., Jhingran, A., Kanungo, T., Rajagopalan, S., Tomkins, A., Tomlin, J., and Zien, J., "Semtag and seeker: Bootstrapping the semantic web via automated semantic annotation," Jan. 01 2003.

[43] E. Buyko, J. Wermter, M. P. . U. H., "Automatically adapting an nlp core engine to the biology domain," in *In Proceedings of the Joint BioLINK-Bio-Ontologies Meeting, Fortaleza, Brasil*, pp. 65–68, 2006.

[44] Elkin, N., "How america searches: Health and wellness.," *iCrossing, a digital company*, pp. 1–17, 2008.

[45] Etzioni, O., Cafarella, M. J., Downey, D., Popescu, A.-M., Shaked, T., Soderland, S., Weld, D. S., and Yates, A., "Methods for domain-independent information extraction from the web: An experimental comparison," in *AAAI* (McGuinness, D. L. and Ferguson, G., eds.), pp. 391–398, AAAI Press / The MIT Press, 2004.

[46] Evans, B. M. and Chi, E. H., "Towards a model of understanding social search," in *CSCW '08: Proceedings of the ACM 2008 conference on Computer supported cooperative work*, (New York, NY, USA), pp. 485–494, ACM, 2008.

[47] Fellbaum, *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*. The MIT Press, May 1998.

[48] Ferrucci, D. and Lally, A., "Building an example application with the unstructured information management architecture," *IBM Systems Journal*, vol. 43, no. 3, pp. 455–455–475, 2004. Systems design; INODISY0001205111; ISY; 2004; 222418995; Lally, A; Information management; Studies; 455-475; English; Armonk; IBMSA7; 26490; 13651391; 708283101; Software engineering; Ferrucci, D; IBM Corp; Copyright International Business Machines Corporation 2004.

[49] Fox, E. A., Hix, D., Nowell, L. T., Brueni, D. J., Rao, D., Wake, W. C., and Heath, L. S., "Users, user interfaces, and objects: Envision, a digital library," *J. Am. Soc. Inf. Sci.*, vol. 44, no. 8, pp. 480–491, 1993.

[50] Francis, Anthony G., J., *Context-sensitive asynchronous memory : a general experience-based method for managing information access in cognitive agents*. PhD thesis, Georgia Institute of Technology, 2000.

[51] Fuchs, D., Fuchs, L. S., Mathes, P. G., and Simmons, D. C., "Peer-assisted learning strategies: Making classrooms more responsive to diversity," *American Educational Research Journal*, vol. 34, no. 1, p. 174, 1997.

[52] Fukuda, K., Tsunoda, T., Tamura, A., and Takagi, T., "Toward information extraction: Identifying protein names from biological papers," Sept. 26 1998.

[53] Gildea, D. and Jurafsky, D., "Automatic labeling of semantic roles," *Computational Linguistics*, vol. 28, no. 3, pp. 245–288, 2002.

[54] Girju, R. and Moldovan, D. I., "Text mining for causal relations," in *FLAIRS Conference* (Haller, S. M. and Simmons, G., eds.), pp. 360–364, AAAI Press, 2002.

[55] Haghighi, A., Ng, A. Y., and Manning, C. D., "Robust textual inference via graph matching," in *HLT/EMNLP*, The Association for Computational Linguistics, 2005.

[56] Hearst, M. A., "Automatic acquisition of hyponyms from large text corpora," in *COLING*, pp. 539–545, 1992.

[57] "Google Stays at 72 Percent of U.S. Searches in February 2009." News Release, March 2009. [online] `http://press.experian.com/documents/showdoc.cfm?doc=3455`.

[58] Horowitz, D. and Kamvar, S. D., "The anatomy of a large-scale social search engine," in *WWW*, 2010.

[59] Huberman, B. A., Pirolli, P. L. T., Pitkow, J. E., and Lukose, R. M., "Strong regularities in World Wide Web surfing," *Science*, vol. 280, pp. 95–97, Apr. 1998.

[60] Jagadeesh, J., Pingali, P., and Varma, V., "A relevance-based language modeling approach to DUC 2005," 2005.

[61] Johnson-Laird, P., *The computer and the mind*. Harvard Univ. Press, 1988.

[62] Kalfoglou, Y. and Schorlemmer, M., "Ontology mapping: The state of the art," in *Semantic Interoperability and Integration* (Kalfoglou, Y., Schorlemmer, M., Sheth, A., Staab, S., and Uschold, M., eds.), no. 04391 in Dagstuhl Seminar Proceedings, 2005.

[63] Karasavvidis, I., "Distributed Cognition and Educational Practice.," *Journal of Interactive Learning Research*, pp. 11–29, 2002.

[64] Kautz, H., Selman, B., and Shah, M., "Referral Web: combining social networks and collaborative filtering," *Commun. ACM*, vol. 40, pp. 63–65, March 1997.

[65] "The state of search, kelton research whitepaper.." Report, May 2007. [online] `http://searchengineland.com/071023-093541.php`.

[66] Kollock, P. and Smith, M., "Managing the Virtual Commons: Cooperation and Conflict in Computer Communities," pp. 109–128, Amsterdam: John Benjamins, 1996.

[67] Kolodner, J., *Case-based reasoning*. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA, 1993.

[68] Konstan, J. A., Miller, B. N., Maltz, D., Herlocker, J. L., Gordon, L. R., Riedl, J., and Volume, H., "Grouplens: Applying collaborative filtering to usenet news," *Communications of the ACM*, vol. 40, pp. 77–87, 1997.

[69] Krulwich, B. and Burkey, C., "Contactfinder agent: answering bulletin board questions with referrals," in *The 1996 13th National Conference on Artificial Intelligence, AAAI 96. Part 1 (of 2)*, pp. 10–15, 1996.

[70] Kumaran, G. and Allan, J., "A Case for Shorter Queries, and Helping Users Create Them," in *In NAACL-HLT*, pp. 220–227, 2007.

[71] Lampert, A., Dale, R., and Paris, C., "Classifying speech acts using verbal response modes," 2006.

[72] Lassila, O. and Swick, R., "Resource description framework (rdf) model and syntax specification."

[73] Leake, D., *Case-based reasoning: Experiences, lessons and future directions*. MIT Press Cambridge, MA, USA, 1996.

[74] Lee, B., Hendler, T., and Lassila, J., "The semantic web," *Scientific American*, May 2001.

[75] Leskovec, J., Grobelnik, M., and Milic-Frayling, N., "Learning substructures of document semantic graphs for document summarization," 2004.

[76] McCarthy, K., McGinty, L., Smyth, B., and Salamó, M., "The needs of the many: A case-based group recommender system," *Advances in Case-Based Reasoning*, vol. 4106, pp. 196–210, 2006.

[77] McDonald, R., "Extracting relations from unstructured text.," tech. rep., Department of Computer and Information Science, University of Pennsylvania, 2005.

[78] Mott, B. W., Lester, J. C., and Branting, K., "The role of syntactic analysis in textual case retrieval," in *ICCBR Workshops*, pp. 120–127, 2005.

[79] Mukherjea, S. and Sahay, S., "Discovering biomedical relations utilizing the world-wide web," in *Pacific Symposium on Biocomputing* (Altman, R. B., Murray, T., Klein, T. E., Dunker, A. K., and Hunter, L., eds.), pp. 164–175, World Scientific, 2006.

[80] Nah, S., "Connecting social capital offline and online: The effects of internet uses on civic community engagement," in *The American Association for (AAPOR) 59th Annual Conference, 2004 and WAPOR 57th Annual Conference, 2004*, 2004.

[81] Newell, A., *Unified theories of cognition*. Cambridge, MA, USA: Harvard University Press, 1990.

[82] Pazzani, M. J. and Billsus, D., "Content-based recommendation systems," *The adaptive web: methods and strategies of web personalization*, pp. 325–341, 2007.

[83] Peirce, C. S., "The aristotelian syllogistic," in *Collected Papers: Elements of Logic* (Hartshorne, C. and Weiss, P., eds.), pp. 273–283, Cambridge: Harvard University Press, 1965.

[84] Pisanelli, D. and Gangemi, A., "If ontology is the solution, what is the problem," *Ontologies in Medicine*, p. 1, 2004.

[85] Pisanelli, D., *Ontologies in medicine*. IOS Press, 2004.

[86] Pradhan, S., Ward, W., Hacioglu, K., Martin, J., and Jurafsky, D., "Semantic role labeling using different syntactic views," in *ACL*, 2005.

[87] Pradhan, S., Ward, W., Hacioglu, K., and Martin, J. H., "Shallow semantic parsing using support vector machines," 2004.

[88] Ram, A. and Francis, A., "Multi-plan retrieval and adaptation in an experience-based agent," *Case-Based Reasoning: experiences, lessons, and future directions*, pp. 167–184, 1996.

[89] Ram, A., "Interest-based information filtering and extraction in natural language understanding systems," in *In Proceedings of the Bellcore Workshop on High Performance Information Filtering*, 1991.

[90] Richardson, M. and White", R. W., "Supporting synchronous social qna throughout the question lifecycle.," in *WWW'11*, 2011.

[91] Rissland, E. L. and Daniels, J. J., "The synergistic application of cbr to ir," *Artif. Intell. Rev.*, vol. 10, no. 5-6, pp. 441–475, 1996.

[92] Rojo, A., *Participation in scholarly electronic forums*. PhD thesis, University of Toronto, 1995.

[93] Sahay, S., Mukherjea, S., Agichtein, E., Garcia, E. V., Navathe, S. B., and Ram, A., "Discovering semantic biomedical relations utilizing the web," *TKDD*, vol. 2, no. 1, 2008.

[94] Sahay, S., Mukherjea, S., Agichtein, E., Garcia, E. V., Navathe, S. B., and Ram, A., "Discovering semantic biomedical relations utilizing the web," *ACM Trans. Knowl. Discov. Data*, vol. 2, pp. 3:1–3:15, April 2008.

[95] Salton, G., *The SMART Retrieval System—Experiments in Automatic Document Processing*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1971.

[96] Sarasohn-Kahn, J., "The wisdom of patients: Health care meets online social media," *California HealthCare Foundation*, Apr. 2008.

[97] Saunders, C. S., Robey, D., and Vaverek, K. A., "The persistence of status differentials in computer conferencing," *Human Communication Research*, vol. 20, no. 4, pp. 443–472, 1994.

[98] Schank, R., "Conceptual dependency: A theory of natural language understanding.," *Cognitive psychology*, vol. 3, no. 4, pp. 552–631, 1972.

[99] Schank, R. and Abelson, R., "Scripts, plans and knowledge," in *Proceedings of the Fourth International Joint Conference on Artificial Intelligence*, pp. 151–157, 1975.

[100] Seco, N., Veale, T., and Hayes, J., "An intrinsic information content metric for semantic similarity in wordnet," 2004.

[101] Shaban, K. B., Basir, O. A., and Kamel, M., "Document mining based on semantic understanding of text," in *CIARP* (Trinidad, J. F. M., Carrasco-Ochoa, J. A., and Kittler, J., eds.), vol. 4225 of *Lecture Notes in Computer Science*, pp. 834–843, Springer, 2006.

[102] Smith, M., Giraud-Carrier, C., and Purser, N., "Implicit affinity networks and social capital," *Information Technology and Management*, vol. 10, pp. 123–134, Sept. 2009.

[103] Smyth, B., Briggs, P., Coyle, M., and O'Mahony, M. P., "A case-based perspective on social web search," in *Proceedings of the 8th International Conference on Case-Based Reasoning: Case-Based Reasoning Research and Development*, (Berlin, Heidelberg), pp. 494–508, Springer-Verlag, 2009.

[104] Sowa, J., "Semantic networks," *Encyclopedia of Artificial Intelligence*, 1992.

[105] Sowa, J. F., "Conceptual graphs for a data base interface," *IBM Journal of Research and Development*, vol. 20, no. 4, pp. 336–357, 1976.

[106] Spink, A., Jansen, B. J., Wolfram, D., and Saracevic, T., "From e-sex to e-commerce: Web search changes," *Computer*, vol. 35, pp. 107–109, March 2002.

[107] Subramaniam, L. V., Mukherjea, S., Kankar, P., Srivastava, B., Batra, V. S., Kamesam, P. V., and Kothari, R., "Information extraction from biomedical literature: methodology, evaluation and an application," in *Proceedings of the 2003 ACM CIKM International Conference on Information and Knowledge Management (CIKM-03)*, (New York), pp. 410–417, ACM Press, Nov. 2–8 2003.

[108] Tanenblatt, M. A., Coden, A., and Sominsky, I. L., "The conceptmapper approach to named entity recognition.," in *LREC'10*, pp. –1–1, 2010.

[109] Toutanova, K., Haghighi, A., and Manning, C. D., "Joint learning improves semantic role labeling," in *ACL*, 2005.

[110] Vinay, V., Milic-frayling, I. J. C. N., and Wood, K., "On ranking the effectiveness of searches," in *In: Proc. of the 29th Annual Intl ACM SIGIR Conf. on Research and Development in Information Retrieval*, pp. 398–404, ACM Press, 2006.

[111] Vygotsky, L. and Cole, M., *Mind in society: the development of higher psychological processes*. Harvard University Press, 1978.

[112] Webb, G. I., Pazzani, M. J., and Billsus, D., "Machine learning for user modeling," *User Modeling and User-Adapted Interaction*, vol. 11, pp. 19–29, March 2001.

[113] Weber, R., Aha, D., Sandhu, N., and Munoz-Avila, H., "A textual case-based reasoning framework for knowledge management applications," 2001.

[114] Weisband, Suzanne P., S. S. K. and Connolly, T., "Computer mediated communication and social information: Status salience and status differences.," *Academy of Management Journal*, vol. 38, no. 4, pp. 1124–1151, 1995.

[115] Wong, L., "Pies, a protein interaction extraction system," in *Pacific Symposium on Biocomputing* (Altman, R. B., Dunker, A. K., Hunter, L., and Klein, T. E., eds.), vol. 6, (Singapore), pp. 520–531, World Scientific Press, 2001.

[116] Woods, W. A., "Transition network grammars for natural language analysis," *Commun. ACM*, vol. 13, pp. 591–606, October 1970.

[117] Woods, W. A., *Semantics and quantification in natural language question answering*, pp. 205–248. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1986.

[118] Yang, Y., Lad, A., Lao, N., Harpale, A., Kisiel, B., and Rogati, M., "Utility-based information distillation over temporally sequenced documents," in *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '07, (New York, NY, USA), pp. 31–38, ACM, 2007.

[119] Ybarra, O., Burnstein, E., Winkielman, P., Keller, M. C., Manis, M., Chan, E., and Rodriguez, J., "Mental Exercising Through Simple Socializing: Social Interaction Promotes General Cognitive Functioning," *Pers Soc Psychol Bull*, vol. 34, no. 2, pp. 248–259, 2008.

[120] Zelenko, D., Aone, C., and Richardella, A., "Kernel methods for relation extraction," *Journal of Machine Learning Research*, vol. 3, pp. 1083–1106, 2003.

VITA

Saurav's research interest lies broadly in the field of intelligent information access, primarily at the intersection of language processing, user modeling and web based socio-informatics systems. He has previously worked on IBM DeepQA Watson Question Answering system and other health informatics projects at IBM Research Labs and was a finalist in the world wide IBM PhD Fellowship competition. He also co-chaired the CBR Startups workshop at ICCBR 2010.

Related Publications:

Ashwin Ram, Hua Ai, Preetha Ram, Saurav Sahay. Open Social Learning Communities. International Conference on Web Intelligence, Mining and Semantics, WIMS 2011.

Saurav Sahay, Hua Ai and Ashwin Ram. Intentional analysis of medical conversations for community engagement. Flairs 2011

Saurav Sahay and Ashwin Ram. Socio-Semantic Health Information Access. AAAI Spring Symposium 2011.

Saurav Sahay, Stephanie Ahn, Szu-Chia Lu, Brian Sherwell, Anushree Venkatesh and Ashwin Ram. Healthbuzz: Contextual Social Search and Conversations. Third Annual Workshop on Search in Social Media, SSM 2010, New york, USA.

Saurav Sahay, Ashwin Ram. Conversational Framework for Web Search and Recommendations. 'Reasoning from Experiences on the Web' Workshop at International Conference on Case based Reasoning. ICCBR 10

Saurav Sahay, Anushree Venkatesh, Ashwin Ram. Cobot: Real Time Multi User Conversational Search and Recommendations. ACM RecSys 2009 Workshop on Recommender Systems and The Social Web.

Saurav Sahay, Anushree Venkatesh, Ashwin Ram. Collaborative Information Access: A Conversational Search Approach. 'Reasoning from Experiences on the Web' Workshop at 8th International Conference on Case based Reasoning. ICCBR 09

Saurav Sahay, Sougata Mukherjea, Eugene Agichtein, Ernest V Garcia, Shamkant Navathe, Ashwin Ram. Discovering Semantic Biomedical Relations utilizing the Web. ACM Transactions on Knowledge Discovery from Data, 2(1):3, 2008

Saurav Sahay, Sundaresan Venkatasubramanian, Anushree Venkatesh, Priyanka Prabhu, Bharat Ravisekar, Ashwin Ram. iReMedI - Intelligent Retrieval from Medical Information. ECCBR 08.

Saurav Sahay, Eugene Agichtein, Baoli Li, Ernest V Garcia, Ashwin Ram.(2007) Semantic Annotation and Inference for Medical Knowledge Discovery. 2007 NSF Next Generation Data Mining(NGDM) Symposium presentation, October 2007 Baltimore.

Saurav Sahay, Baoli Li, Ernest V Garcia, Eugene Agichtein, Ashwin Ram.(2007) Domain Ontology construction from Biomedical Text. ICAI 2007

Baoli Li, Saurav Sahay, Shreekanth Karvaje, Bharat Ravisekar, Joseph Irwin, Neha Sugandh, Cesar Santana, Eugene Agichtein, Ashwin Ram, Ernest V Garcia. Locating Applicable Knowledge Sentences in Medical Literature. The 6th Georgia Tech-ORNL International Conference on Bioinformatics poster presentation, 2007.

Shreekanth Karvaje, Bharat Ravisekar, Saurav Sahay, Baoli Li, Ernest Garcia, Ashwin Ram.(2007) Discovering Causal Sentences with Automatically Learned Patterns, ISBRA 07 Poster

Sougata Mukherjea, Saurav Sahay (2006) Discovering Biomedical relations utilizing the world-wide web. Pacific Symposium on Biocomputing 2006

N. Polavarapu, S. B. Navathe, R. Ramnarayanan, A. Haque, S. Sahay, Ying Liu. (2005) Investigation into Biomedical Literature Classification using Support Vector Machines. Proceedings of 2005 IEEE Computational Systems Bioinformatics Conference (CSB2005), Stanford University, August 8-11, 2005.